Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 389 (2007) 190–218
Theoretical Computer Science
www.elsevier.com/locate/tcs

Learning languages from positive data and a limited number of short counterexamples

Sanjay Jain[a,*], Efim Kinber[b]

[a] *School of Computing, National University of Singapore, Singapore 117590, Singapore*
[b] *Department of Computer Science, Sacred Heart University, Fairfield, CT 06432-1000, USA*

Received 17 July 2006; received in revised form 10 July 2007; accepted 23 August 2007

Communicated by O. Watanabe

Abstract

We consider two variants of a model for learning languages in the limit from positive data and a limited number of short negative counterexamples (counterexamples are considered to be short if they are smaller than the largest element of input seen so far). Negative counterexamples to a conjecture are examples which belong to the conjectured language but do not belong to the input language. Within this framework, we explore how/when learners using n short (arbitrary) negative counterexamples can be simulated (or simulate) using least short counterexamples or just 'no' answers from a teacher. We also study how a limited number of short counterexamples fairs against unconstrained counterexamples, and also compare their capabilities with the data that can be obtained from *subset*, *superset*, and *equivalence* queries (possibly with counterexamples). A surprising result is that just one short counterexample can sometimes be more useful than any bounded number of counterexamples of arbitrary sizes. Most of the results exhibit salient examples of languages learnable or not learnable within corresponding variants of our models.

Keywords: Inductive inference; Learning in the limit; Positive data; Counterexamples

1. Introduction

Our goal in this paper is to explore how limited amount of negative data, relatively easily available from a teacher, can help learning languages in the limit. There is a long tradition of using two popular different paradigms for exploring learning languages in the limit. One paradigm, learning languages from full positive data (all correct statements of the language), was introduced by Gold in his classical paper [9]. In this model, **TxtEx**, the learner stabilizes in the limit to a grammar generating the target language. In another popular variant of this model, **TxtBc**, defined in [5] and [17] (see also [2] and [6]), almost all conjectures outputted by the learner are correct grammars describing the target language. The second popular paradigm, learning using queries to a teacher (oracle) was introduced by Angluin in [1]. In particular, Angluin considered three types of queries: subset, superset, and

* Corresponding author. Tel.: +65 6874 7842; fax: +65 6779 4580.
E-mail addresses: sanjay@comp.nus.edu.sg (S. Jain), kinbere@sacredheart.edu (E. Kinber).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2007.08.010

equivalence queries – when a learner asks if a current hypothesis generates a subset or a superset of the target language, or, respectively, generates exactly the target language. If the answer is negative, the teacher may provide a *counterexample* showing where the current hypothesis errs. This model has been used for exploring language learning primarily in the situation when no data is available in advance (see, for example, [14,13]). In [11], the two models were combined together: a learner gets full positive data and can query the teacher if the current conjecture is correct. On one hand, this model reflects the fact that a learner, during a process of acquisition of a new language, potentially gets access to all correct statements. On the other hand, this model adds another important tool, typically available, say, to a child learning a new language: a possibility to communicate with a teacher. Sometimes, this possibility may be really vital for successful learning. For example, if a learner of English past tense, having received on the input "call – called", "fall – fell", infers the rule implying that both past tense forms "called, cell" and "falled, fell" are possible, then this rule can be refuted only by counterexamples from a teacher.

In this context, subset queries are of primary interest, as they provide *negative counterexamples* if the learner errs, while other types of queries may provide positive 'counterexamples' eventually available on the input anyway (still, as it was shown in [10], the sequel paper to [11], superset and equivalence queries can make some difference even in presence of full positive data). Consequently, one can consider the learner for **NCEx**-model as defined in [11] (and its variant **NCBc** corresponding to **TxtBc** – **NC** here stands for 'negative counterexamples'), as making a subset query for each of its conjectures. When a learner tests every conjecture, potentially he/she can get indefinite number of counterexamples (still this number is, of course, finite if the learner learns the target language in the limit correctly). In [10] the authors explored learning from positive data and *bounded* amount of additional negative data. In this context, one can consider three different scenarios of how subset queries and corresponding negative counterexamples (if any) can be used:

– only a bounded number (up to n) of subset queries is allowed during the learning process; this model was considered in [10] under the name $\mathbf{SubQ}^n$;
– the learner makes subset query for every conjecture until n negative answers have been received; that is, the learner can ask potentially indefinite number of questions (however, still finite if the learning process eventually gives a correct grammar), but he is *charged* only when receiving a negative answer; this model was considered in [10] under the name $\mathbf{NC}^n$;
– the learner makes subset queries for conjectures, when deemed necessary, until n negative answers have been received; in the sequel, we will refer to this model as $\mathbf{GNC}^n$, where **GNC** denotes 'generalized model of learning via negative counterexamples'.

Note that the $\mathbf{GNC}^n$-model combines the features of the first two (we will also demonstrate that it is stronger than each of the first two).

All three models $\mathbf{SubQ}^n$, $\mathbf{NC}^n$, and $\mathbf{GNC}^n$ provide certain complexity measure (in the spirit of [8]) for learning languages that cannot be learned from positive data alone.

Negative counterexamples provided by the teacher in all these models are of arbitrary size. Some researchers in the field considered other types of negative data available for learners from full positive data. For example, negative data provided to learners in the model considered in [4] is preselected – in this situation just a very small amount of negative data can greatly enhance learning capabilities. A similar model was considered in [15].

In this paper we consider models $\mathbf{SubQ}^n$, $\mathbf{NC}^n$, and $\mathbf{GNC}^n$ when the teacher provides a negative counterexample only if there is one whose size does not exceed the size of the longest statement seen so far. While learning from full positive data and negative counterexamples of arbitrary size can be interesting and insightful on its own right, providing arbitrary examples immediately (as it is assumed in the models under consideration) may be somewhat unrealistic – in fact, it may significantly slow down the learning process, if not making it impossible. On the other hand, it is reasonable to assume that the teacher can reasonably quickly provide a counterexample (if any), if its size is bounded by the largest statement in the input seen so far. Following notations in [10], we denote corresponding variants of our three models by $\mathbf{BSubQ}^n$, $\mathbf{BNC}^n$, and $\mathbf{BGNC}^n$, respectively. We also consider the following two variants of the above model.

(i) The least counterexample (if any) is provided rather than an arbitrary one (these variants are denoted by adding a prefix **L** as in $\mathbf{LBSubQ}^n$, $\mathbf{LBNC}^n$ and $\mathbf{LBGNC}^n$).

(ii) Following [1] and [10] we consider the case when the teacher, responding to a query, answers just 'no' if a counterexample of the size not exceeding the size of the largest statement seen so far exists – not providing the actual example; otherwise, the teacher answers 'yes' (we add a prefix **Res** to the name of a model to denote this variant).

It must be noted that, as it is shown in [10], $\mathbf{BSubQ}^n$ does not provide any advantages over learning just from positive data. Therefore, we concentrate on $\mathbf{BNC}^n$, $\mathbf{BGNC}^n$ and their **L** and **Res**-variants.

In this paper we will explore relationships between $\mathbf{BNC}^n$, $\mathbf{BGNC}^n$, $\mathbf{NC}^n$ and $\mathbf{GNC}^n$ (as well as their **L** and **Res**-variants). In this context, we, in particular, demonstrate advantages that our **B**-variants of learning (even **ResB**) can have over $\mathbf{GNC}^n$ in terms of the number of mind changes needed to arrive at the right conjecture. We consider also learning with bounded number of two other types of queries, superset and equivalence, and discuss how their capabilities in the presence of full positive data fair against **B** and **ResB** types of learning with bounded numbers of counterexamples/'no' – answers (as it was noted above, even though superset and equivalence queries may provide positive 'counterexamples', there are circumstances when this can help even in the presence of full positive data – see, for example, Theorems 19 and 22 in [10]).

Most of our results provide salient examples of classes learnable (or not learnable) within corresponding models.

The paper has the following structure. In Section 2 we introduce necessary notations and definitions needed in the rest of the paper. In particular, we define some variants of the classical Gold's model of learning from texts (positive data): **TxtEx** – when the learner stabilizes to a correct (or nearly correct) conjecture generating the target language, and **TxtBc** – its behaviorally correct counterpart. In Section 3 we define learnability from positive data via uniformly bounded number of queries to the teacher (oracle). In particular, we define learning via queries returning the least one or no counterexamples (just the answers 'yes' or 'no' in the latter case). In all these models the learning algorithm is *charged* for every query that it makes. This section also gives the reader general understanding of how learning from positive data via subset queries works.

In Section 4, for both major models of learnability in the limit, **TxtEx** and **TxtBc**, we define two variants of learning from positive data and a uniformly bounded number of counterexamples: $\mathbf{NC}^n$ and $\mathbf{GNC}^n$, where the learner makes subset queries and is charged for every negative answer from a teacher (rather than for every query, as in the query model in Section 3). We then define the main models considered in this paper: $\mathbf{BNC}^n$ and $\mathbf{BGNC}^n$, as well as **Res** variants of both. We also formally define the **L** variant for all these models. In addition, we establish some useful facts regarding the model **GNC**, as it is introduced in this paper for the first time.

In Section 5 we explore relationships between different bounded negative counterexample models. In particular, we study the following two problems: under which circumstances, (a) **B**-learners receiving just answers 'yes' or 'no' can simulate the learners receiving short (possibly, even least) counterexamples; (b) learners receiving arbitrary short counterexamples can simulate the ones receiving the least short counterexamples. First, we note that in all the variants of the paradigms **TxtEx** and **TxtBc**, an $\mathbf{LBNC}^n$-learner can be always simulated by a $\mathbf{ResBNC}^{2n-1}$-learner: $2n-1$ 'no' answers are enough to simulate n explicit negative counterexamples (similar fact holds also for the $\mathbf{LBGNC}^n$-learners). Moreover, for the $\mathbf{Bc}^*$-type of learnability (when almost all conjectures contain any finite number of errors), the number $2n-1$ in the above result drops to n (Theorem 27; note that, for learning via limited number of arbitrary or least counterexamples, the number $2n-1$ could not be lowered even for $\mathbf{Bc}^*$-learners, as shown in [10]). On the other hand, the number $2n-1$ of negative answers/counterexamples cannot be lowered for the learning types $\mathbf{Ex}^*$ (when any finite number of errors in the limiting correct conjecture) and $\mathbf{Bc}^m$ (when the number of errors in almost all conjectures is uniformly bounded by some m) for both tasks (a) and (b). Namely, there exist $\mathbf{LBNC}^n\mathbf{Ex}$-learnable classes of languages that cannot be learned by $\mathbf{BGNC}^{2n-2}\mathbf{Bc}^m$ or $\mathbf{BGNC}^{2n-2}\mathbf{Ex}^*$-learners (Theorem 24) and there exist $\mathbf{BNC}^n\mathbf{Ex}$-learnable classes that cannot be learned by $\mathbf{ResBNC}^{2n-2}\mathbf{Bc}^m$ or $\mathbf{ResBNC}^{2n-2}\mathbf{Ex}^*$-learners (Theorems 25 and 26). We also show that a $\mathbf{LBNCEx}^*$-learner can be always simulated by a **ResBNCBc**-learner – when the number of negative answers/counterexamples is unbounded.

In Section 6 we explore relationships between our models when the counterexamples considered are short or unconstrained. First, we demonstrate how short counterexamples can be of advantage over unconstrained ones while learning from positive data and a bounded number of counterexamples. A somewhat surprising result is that sometimes one 'no' answer, just indicating that a short counterexample exists, can do more than any number n of arbitrary (or even least) counterexamples used by (the strongest) $\mathbf{LGNC}^n\mathbf{Bc}^*$-learners (Theorem 33). Note that the advantages of least examples/counterexamples in speeding up learning have been studied in other situations also, such as learning of non-erasing pattern languages [19]. However, in our model of **BNC**-learning versus **LNC**-learning, the **LNC**-learner does get least counterexamples, and **BNC**-learner gets just a counterexample, if there exists one below the largest positive data seen so far. This seems on the surface to hurt, as **BNC**-learner is likely to get less (negative) data. In fact, that is the case when we do not bound the number of counterexamples received. However, when we consider counting/bounding, there is a *charge* for every counterexample. Consequently, a **BNC**-learner is not being charged for (unnecessary) negative data, if it does not receive it. As a result, the possibility of getting negative data which are $\leq$ largest positive data seen in the input so far can be turned to an advantage – in terms of cost of learning. This is what is exploited in getting this result. We also show that sometimes a $\mathbf{ResBNC}^1\mathbf{Ex}$-learner can use just one mind change (and one 'no' answer witnessing existence of a short counterexample) to learn classes of languages not learnable by any **GNCEx**-learner using any bounded number of mind changes and an unbounded (finite) number of arbitrary counterexamples (Theorem 35). On the other hand, least counterexamples used by **NC**-type learners make a difference: any **LBNCEx**-learner using at most m mind changes and any (unbounded) number of counterexamples can be simulated by a $\mathbf{LNC}^m$-learner using at most m mind changes and at most m least counterexamples.

In Section 7 we study how learning via limited number of short counterexamples fairs against learning via finite number of subset, superset, and equivalence queries (note that, as shown in [10], if answers 'no'/counterexamples to queries are of **B**-type (i.e. constrained to be short), then they do not give any advantage over regular learnability by **TxtEx** or **TxtBc**-learners, thus we consider here only queries returning arbitrary or least counterexamples or just 'no' answers assuming existence of a counterexample). In some cases, just one query, providing only the answer, without associated counterexample, can give one a learning advantage compared to any number n of least short counterexamples used by $\mathbf{BNC}^n\mathbf{Bc}$ or $\mathbf{BGNC}^n\mathbf{Bc}$-learners (sometimes even making errors in almost all correct conjectures). On the other hand $\mathbf{Bc}^m$ and $\mathbf{Ex}^*$-learners using any finite number of superset queries can be simulated by **ResBNCBc**-learners making just one error in almost all correct conjectures if an unbounded number of 'no' answers is allowed (Theorem 42). Conversely, one restricted 'no' answer (just assuming existence of a short counterexample) can sometimes do better than any (bounded) number of queries of any type while getting least counterexamples.

We hope that our models and results shed a new light on how limited negative data can help learning languages in the limit.

2. Notations and preliminaries

Any unexplained recursion theoretic notation is from [18]. The symbol N denotes the set of natural numbers, $\{0, 1, 2, 3, \ldots\}$. Symbols $\emptyset$, $\subseteq$, $\subset$, $\supseteq$, and $\supset$ denote empty set, subset, proper subset, superset, and proper superset, respectively. $D_0, D_1, \ldots,$ denotes a canonical recursive indexing of all the finite sets [18, Page 70]. We assume that if $D_i \subseteq D_j$ then $i \leq j$ (the canonical indexing defined in [18] satisfies this property). Cardinality of a set S is denoted by $\mathrm{card}(S)$. I_m denotes the set $\{x \mid x \leq m\}$. The maximum and minimum of a set are denoted by $\max(\cdot)$, $\min(\cdot)$, respectively, where $\max(\emptyset) = 0$ and $\min(\emptyset) = \infty$. $L_1 \mathbf{\Delta} L_2$ denotes the symmetric difference of L_1 and L_2, that is $L_1 \mathbf{\Delta} L_2 = (L_1 - L_2) \cup (L_2 - L_1)$. For a natural number a, we say that $L_1 =^a L_2$, iff $\mathrm{card}(L_1 \mathbf{\Delta} L_2) \leq a$. We say that $L_1 =^* L_2$, iff $\mathrm{card}(L_1 \mathbf{\Delta} L_2) < \infty$. Thus, we take $n < * < \infty$, for all $n \in N$. If $L_1 =^a L_2$, then we say that L_1 is an a-variant of L_2.

We let $\langle \cdot, \cdot \rangle$ stand for an arbitrary, computable, bijective mapping from $N \times N$ onto N [18]. We assume without loss of generality that $\langle \cdot, \cdot \rangle$ is monotonically increasing in both of its arguments. We define $\pi_1(\langle x, y \rangle) = x$ and $\pi_2(\langle x, y \rangle) = y$. We can extend pairing function to multiple arguments by using $\langle i_1, i_2, \ldots, i_k \rangle = \langle i_1, \langle i_2, \langle \ldots, \langle i_{k-1}, i_k \rangle \rangle \rangle \rangle$.

We let $\{W_i\}_{i \in N}$ denote an acceptable numbering of all r.e. sets. Symbol $\mathcal{E}$ will denote the set of all r.e. languages. Symbol L, with or without decorations, ranges over $\mathcal{E}$. By $\overline{L}$, we denote the complement of L, that is $N - L$. Symbol $\mathcal{L}$, with or without decorations, ranges over subsets of $\mathcal{E}$. By $W_{i,s}$ we denote the set W_i enumerated within s steps, in some standard computable method of enumerating W_i.

We now present concepts from language learning theory. The next definition introduces the concept of a *sequence* of data.

Definition 1. (a) A *sequence* σ is a mapping from an initial segment of N into $(N \cup \{\#\})$. The empty sequence is denoted by Λ.

(b) The *content* of a sequence σ, denoted $\mathrm{content}(\sigma)$, is the set of natural numbers in the range of σ.

(c) The *length* of σ, denoted by $|\sigma|$, is the number of elements in σ. So, $|\Lambda| = 0$.

(d) For $n \leq |\sigma|$, the initial sequence of σ of length n is denoted by $\sigma[n]$. So, $\sigma[0]$ is Λ.

Intuitively, #'s represent pauses in the presentation of data. We let σ, τ, and γ, with or without decorations, range over finite sequences. We denote the sequence formed by the concatenation of τ at the end of σ by $\sigma\diamond\tau$. For ease of notation, we often drop $\diamond$, and just use $\sigma\tau$ to denote concatenation of σ and τ. Sometimes we abuse the notation and use σx to denote the concatenation of sequence σ and the sequence of length 1 which contains the element x. SEQ denotes the set of all finite sequences.

Definition 2 (*[9]*). (a) A *text* T for a language L is a mapping from N into $(N \cup \{\#\})$ such that L is the set of natural numbers in the range of T. $T(i)$ represents the $(i+1)$th element in the text.
(b) The *content* of a text T, denoted by content(T), is the set of natural numbers in the range of T; that is, the language which T is a text for.
(c) $T[n]$ denotes the finite initial sequence of T with length n.

Definition 3 (*[9]*). A *language learning machine from texts* is an algorithmic device which computes a mapping from SEQ into N.

We let $\mathbf{M}$, with or without decorations, range over learning machines. $\mathbf{M}(T[n])$ is interpreted as the grammar (index for an accepting program) conjectured by the learning machine $\mathbf{M}$ on the initial sequence $T[n]$. We say that $\mathbf{M}$ converges on T to i, (written: $\mathbf{M}(T)\downarrow = i$) iff $(\forall^\infty n)[\mathbf{M}(T[n]) = i]$.

There are several criteria for a learning machine to be successful on a language. Below we define some of them. All of the criteria defined below are variants of the **Ex**-style and **Bc**-style learning described in the Introduction; in addition, they allow a finite number of errors in almost all conjectures (uniformly bounded, or arbitrary).

Definition 4 (*[9,5]*). Suppose $a \in N \cup \{*\}$.

(a) $\mathbf{M}$ $\mathbf{TxtEx}^a$-*identifies a text* T just in case $(\exists i \mid W_i =^a \text{content}(T))\ (\forall^\infty n)[\mathbf{M}(T[n]) = i]$.
(b) $\mathbf{M}$ $\mathbf{TxtEx}^a$-*identifies an r.e. language* L (written: $L \in \mathbf{TxtEx}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies each text for L.
(c) $\mathbf{M}$ $\mathbf{TxtEx}^a$-*identifies a class* $\mathcal{L}$ of r.e. languages (written: $\mathcal{L} \subseteq \mathbf{TxtEx}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies each language from $\mathcal{L}$.
(d) $\mathbf{TxtEx}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtEx}^a(\mathbf{M})]\}$.

Definition 5 (*[5]*). Suppose $a \in N \cup \{*\}$.

(a) $\mathbf{M}$ $\mathbf{TxtBc}^a$-*identifies a text* T just in case $(\forall^\infty n)[W_{\mathbf{M}(T[n])} =^a L]$.
(b) $\mathbf{M}$ $\mathbf{TxtBc}^a$-*identifies an r.e. language* L (written: $L \in \mathbf{TxtBc}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{TxtBc}^a$-identifies each text for L.
(c) $\mathbf{M}$ $\mathbf{TxtBc}^a$-*identifies a class* $\mathcal{L}$ of r.e. languages (written: $\mathcal{L} \subseteq \mathbf{TxtBc}^a(\mathbf{M})$) just in case $\mathbf{M}$ $\mathbf{TxtBc}^a$-identifies each language from $\mathcal{L}$.
(d) $\mathbf{TxtBc}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtBc}^a(\mathbf{M})]\}$.

For $a = 0$, we often write **TxtEx** and **TxtBc**, instead of $\mathbf{TxtEx}^0$ and $\mathbf{TxtBc}^0$, respectively.

Definition 6 (*[7]*). σ is said to be a **TxtEx**-*stabilizing sequence* for $\mathbf{M}$ on L, iff (i) content$(\sigma) \subseteq L$, and (ii) for all σ' such that $\sigma \subseteq \sigma'$ and content$(\sigma') \subseteq L$, $\mathbf{M}(\sigma) = \mathbf{M}(\sigma')$.

Definition 7 (*[3,16]*). For $a \in N \cup \{*\}$, σ is said to be a $\mathbf{TxtEx}^a$-*locking sequence* for $\mathbf{M}$ on L, iff (i) content$(\sigma) \subseteq L$, (ii) for all σ' such that $\sigma \subseteq \sigma'$ and content$(\sigma') \subseteq L$, $\mathbf{M}(\sigma) = \mathbf{M}(\sigma')$, and (iii) $W_{\mathbf{M}(\sigma)} =^a L$.

Theorem 8 (*[3,16]*). *Suppose* $a \in N \cup \{*\}$. *Suppose* $\mathbf{M}$ $\mathbf{TxtEx}^a$-*identifies* $\mathcal{L}$. *Then, for all* $L \in \mathcal{L}$,

(a) *there exists a* $\mathbf{TxtEx}^a$-*locking sequence for* $\mathbf{M}$ *on* L*;*
(b) *for all* σ *such that* content$(\sigma) \subseteq L$, *there exists a* $\mathbf{TxtEx}^a$-*locking sequence, extending* σ, *for* $\mathbf{M}$ *on* L.

Definition 9 (*Based on [3,16]*). For $a \in N \cup \{*\}$, σ is said to be a $\mathbf{TxtBc}^a$-*locking sequence* for $\mathbf{M}$ on L, iff (i) content$(\sigma) \subseteq L$, and (ii) for all σ' such that $\sigma \subseteq \sigma'$ and content$(\sigma') \subseteq L$, $W_{\mathbf{M}(\sigma')} =^a L$.

Theorem 10 (*Based on [3,16]*). *Suppose* $a \in N \cup \{*\}$. *Suppose* $\mathbf{M}$ $\mathbf{TxtBc}^a$-*identifies* $\mathcal{L}$. *Then, for all* $L \in \mathcal{L}$,

(a) *there exists a* $\mathbf{TxtBc}^a$-*locking sequence for* $\mathbf{M}$ *on* L*;*
(b) *for all* σ *such that* content$(\sigma) \subseteq L$, *there exists a* $\mathbf{TxtBc}^a$-*locking sequence, extending* σ, *for* $\mathbf{M}$ *on* L.

Similar stabilizing sequence/locking sequence results can be obtained for the criteria of inference discussed below.

We let INIT $= \{L \mid (\exists i)[L = \{x \mid x \leq i\}]\}$.

For any L, let $cyl(L) = \{\langle i, x\rangle \mid i \in L, x \in N\}$. Let $cyl(\mathcal{L}) = \{cyl(L) \mid L \in \mathcal{L}\}$.

Let CYL_i denote the language $\{\langle i, x\rangle \mid x \in N\}$.

Let **FINITE** denote the class of all finite languages.

The following proposition is useful in proving many of our results.

Proposition 11 ([9]). *Suppose L is an infinite language, $S \subseteq L$, and $L - S$ is infinite. Let $C_0 \subseteq C_1 \subseteq \cdots$ be an infinite sequence of finite sets such that $\bigcup_{i\in N} C_i = L$. Then $\{L\} \cup \{S \cup C_i \mid i \in N\}$ is not in* $\mathbf{TxtBc}^*$.

3. Learning with queries

In this section we define learning with queries. The learning criteria considered in this section are essentially from [10]. The kind of queries [1] considered are

(i) subset queries, i.e., for a queried language Q, 'is $Q \subseteq L$?', where L is the language being learned;
(ii) equivalence queries, i.e., for a queried language Q, 'is $Q = L$?', where L is the language being learned;
(iii) superset queries, i.e., for a queried language Q, 'is $Q \supseteq L$?', where L is the language being learned.

In the model of learning, the learner is allowed to ask queries such as above during its computation. If the answer to query is 'no', we additionally can have the following possibilities:

(a) Learner is given an arbitrary counterexample (for subset query, counterexample is a member of $Q - L$; for equivalence query the counterexample is a member of $L\boldsymbol{\Delta}Q$; for superset query the counterexample is a member of $L - Q$);
(b) Learner is given the least counterexample;
(c) Learner is just given the answer 'no', without any counterexample.

We would often also consider bounds on the number of queries. We first formalize the definition of a learner which uses queries.

Definition 12 ([10]). A learner using queries, can ask a query of the form '$W_j \subseteq L$?' ('$W_j = L$?', '$W_j \supseteq L$?') on any input σ. Answer to the query is 'yes' or 'no' (along with a possible counterexample). Then, based on input σ and answers received for queries made on prefixes of σ, $\mathbf{M}$ outputs a conjecture (from N).

We assume (without loss of generality for the learning criteria considered in this paper) that on any particular input σ, $\mathbf{M}$ asks at most one query. Also note that the queries we allow are for recursively enumerable languages, which are posed to the teacher using a grammar (index in an acceptable numbering of all recursively enumerable languages) for the language.

We now formalize learning via subset queries.

Definition 13 ([10]). Suppose $a \in N \cup \{*\}$.

(a) $\mathbf{M}$ $\mathbf{SubQ}^a\mathbf{Ex}$-*identifies* a language L (written: $L \in \mathbf{SubQ}^a\mathbf{Ex}(\mathbf{M})$) iff for any text T for L, it behaves as follows:
  (i) The number of queries $\mathbf{M}$ asks on prefixes of T is bounded by a (if $a = *$, then the number of such queries is finite). Furthermore, all the queries are of the form '$W_j \subseteq L$?'
  (ii) Suppose the answers to the queries are made as follows. For a query '$W_j \subseteq L$?', the answer is 'yes' if $W_j \subseteq L$, and the answer is 'no' if $W_j - L \neq \emptyset$. For 'no' answers, $\mathbf{M}$ is also provided with a counterexample, $x \in W_j - L$. Then, for some k such that $W_k = L$, for all but finitely many n, $\mathbf{M}(T[n])$ outputs the grammar k.
(b) $\mathbf{M}$ $\mathbf{SubQ}^a\mathbf{Ex}$-*identifies* a class $\mathcal{L}$ of languages (written: $\mathcal{L} \subseteq \mathbf{SubQ}^a\mathbf{Ex}(\mathbf{M})$) iff it $\mathbf{SubQ}^a\mathbf{Ex}$-identifies each $L \in \mathcal{L}$.
(c) $\mathbf{SubQ}^a\mathbf{Ex} = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{SubQ}^a\mathbf{Ex}(\mathbf{M})]\}$.

$\mathbf{LSubQ}^a\mathbf{Ex}$-identification and $\mathbf{ResSubQ}^a\mathbf{Ex}$-identification can be defined similarly, where for $\mathbf{LSubQ}^a\mathbf{Ex}$-identification the learner gets the least counterexample for 'no' answers, and for $\mathbf{ResSubQ}^a\mathbf{Ex}$-identification, the learner does not get any counterexample along with the 'no' answers.

For $a, b \in N \cup \{*\}$, for $\mathbf{I} \in \{\mathbf{Ex}^b, \mathbf{Bc}^b\}$, one can similarly define $\mathbf{SubQ}^a\mathbf{I}$, $\mathbf{SupQ}^a\mathbf{I}$, $\mathbf{EquQ}^a\mathbf{I}$, $\mathbf{LSubQ}^a\mathbf{I}$, $\mathbf{LSupQ}^a\mathbf{I}$, $\mathbf{LEquQ}^a\mathbf{I}$, $\mathbf{ResSubQ}^a\mathbf{I}$, $\mathbf{ResSupQ}^a\mathbf{I}$, and $\mathbf{ResEquQ}^a\mathbf{I}$.

For identification with queries, where there is a bound n on the number of queries asked, we will assume without loss of generality that the learner never asks more than n queries, irrespective of whether the input language belongs to the class being learned, or whether the answers given to earlier queries are correct.

4. Learning with negative counterexamples to conjectures

In this section we define two models of learning languages from positive data and negative counterexamples to conjectures. Both models are based on the general idea of learning from positive data and subset queries for the conjectures.

Intuitively, for learning with negative counterexamples to conjectures, we may consider the learner being provided a text, one element at a time, along with a negative counterexample to the latest conjecture, if any. (One may view this counterexample as a response of the teacher to the subset query when it is tested if the language generated by the conjecture is a subset of the target language). One may model the list of counterexamples as a second text for negative counterexamples being provided to the learner. Thus the learning machines get as input two texts, one for positive data, and other for negative counterexamples.

We say that $\mathbf{M}(T, T')$ converges to a grammar i, iff for all but finitely many n, $\mathbf{M}(T[n], T'[n]) = i$.

First, we define the basic model of learning from positive data and negative counterexamples to conjectures. In this model, if a conjecture contains elements not in the target language, then a counterexample is provided to the learner. **NC** in the definition below stands for 'negative counterexample'.

Definition 14 (*[11]*). Suppose $a \in N \cup \{*\}$.

(a) $\mathbf{M}$ $\mathbf{NCEx}^a$*-identifies a language* L (written: $L \in \mathbf{NCEx}^a(\mathbf{M})$) iff for all texts T for L, and for all T' satisfying the condition:

$$(T'(n) \in S_n,\ \text{if } S_n \neq \emptyset) \text{ and } (T'(n) = \#,\ \text{if } S_n = \emptyset),\ \text{ where } S_n = \overline{L} \cap W_{\mathbf{M}(T[n],T'[n])}$$

$\mathbf{M}(T, T')$ converges to a grammar i such that $W_i =^a L$.

(b) $\mathbf{M}$ $\mathbf{NCEx}^a$*-identifies* a class $\mathcal{L}$ of languages (written: $\mathcal{L} \subseteq \mathbf{NCEx}^a(\mathbf{M})$), iff $\mathbf{M}$ $\mathbf{NCEx}^a$-identifies each language in the class.

(c) $\mathbf{NCEx}^a = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{NCEx}^a(\mathbf{M})]\}$.

For $\mathbf{LNCEx}^a$-criteria of inference, we provide the learner with the least counterexample rather than an arbitrary one. The criteria $\mathbf{LNCEx}^a$ of learning can thus be defined similar to $\mathbf{NCEx}^a$, by requiring ($T'(n) = \min(S_n)$, if $S_n \neq \emptyset$) and ($T'(n) = \#$, if $S_n = \emptyset$) in clause (a) above (instead of $T'(n)$ being an arbitrary member of S_n).

Similarly, one can define $\mathbf{ResNCEx}^a$, where the learner is just told that the latest conjecture is or is not a subset of the input language, but is not provided any counterexamples in the case of 'no' answer.

For $\mathbf{BNCEx}^a$-criteria of inference, we update the definition of S_n in clause (a) of the definition of $\mathbf{NCEx}^a$-identification as follows: $S_n = \overline{L} \cap W_{\mathbf{M}(T[n],T'[n])} \cap \{x \mid x \leq \max(\text{content}(T[n]))\}$.

We can similarly define the criteria of inference $\mathbf{ResBNCEx}^a$ and $\mathbf{LBNCEx}^a$, as well as $\mathbf{NCBc}^a$, $\mathbf{LNCBc}^a$, $\mathbf{ResNCBc}^a$, $\mathbf{BNCBc}^a$, $\mathbf{ResBNCBc}^a$ and $\mathbf{LBNCBc}^a$. We refer the reader to [11] for more details, discussion and results about the various variations of **NCI**-criteria.

Similarly, one can define the models $\mathbf{BSubQ}^a\mathbf{I}$ for the learning via a finite number of subset queries. However, we will not consider these criteria of learning, as they have been shown to be same as **I** in the paper [10].

For $m \in N$, one may also consider the model, $\mathbf{NC}^m\mathbf{I}$, where, for learning a language L, the $\mathbf{NC}^m\mathbf{I}$-learner is provided counterexamples only for its first m conjectures which are not subsets of L. For remaining conjectures, the answer provided is always #. In other words, the learner is 'charged' only for the first m negative counterexamples, and the subset queries for later conjectures are not answered. Following is the formal definition.

Definition 15 (*[10]*). Suppose $a \in N \cup \{*\}$, and $m \in N$.

(a) $\mathbf{M}$ $\mathbf{NC}^m\mathbf{Ex}^a$*-identifies a language* L (written: $L \in \mathbf{NC}^m\mathbf{Ex}^a(\mathbf{M})$) iff for all texts T for L, and for all T' satisfying the condition:

$$(T'(n) \in S_n,\ \text{if } S_n \neq \emptyset \text{ and } \text{card}(\{r \mid r < n \text{ and } T'(r) \neq \#\}) < m);$$

$$(T'(n) = \#, \text{if } S_n = \emptyset \text{ or } \text{card}(\{r \mid r < n \text{ and } T'(r) \neq \#\}) \geq m),\ \text{ where } S_n = \overline{L} \cap W_{\mathbf{M}(T[n],T'[n])}$$

$\mathbf{M}(T, T')$ converges to a grammar i such that $W_i =^a L$.

(b) $\mathbf{M}$ $\mathbf{NC}^m\mathbf{Ex}^a$-*identifies* a class $\mathcal{L}$ of languages (written: $\mathcal{L} \subseteq \mathbf{NC}^m\mathbf{Ex}^a(\mathbf{M})$), iff $\mathbf{M}$ $\mathbf{NC}^m\mathbf{Ex}^a$-identifies each language in the class.
(c) $\mathbf{NC}^m\mathbf{Ex}^a = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{NC}^m\mathbf{Ex}^a(\mathbf{M})]\}$.

For $a \in N \cup \{*\}$ and $\mathbf{I} \in \{\mathbf{Ex}^a, \mathbf{Bc}^a\}$, one can similarly define $\mathbf{BNC}^m\mathbf{I}$, $\mathbf{LBNC}^m\mathbf{I}$, $\mathbf{ResBNC}^m\mathbf{I}$ and $\mathbf{LNC}^m\mathbf{I}$, $\mathbf{ResNC}^m\mathbf{I}$ and $\mathbf{NC}^m\mathbf{Bc}^a$.

GNCI-identification model is same as the model of **NCI**-identification, except that counterexamples are provided to the learner only when it explicitly requests for such via a 'is this conjecture a subset of the target language' question (which we refer to as a conjecture-subset question). This clearly does not make a difference if there is no bound on the number of questions asked resulting in counterexamples. However when there is such a bound, then this may make a difference, as the **GNC**-learner may avoid getting a counterexample on some conjectures by not asking the conjecture-subset question. Thus, we will only deal with **GNC** model when there is a requirement of a bounded number of counterexamples. For $a \in N \cup \{*\}$ and $\mathbf{I} \in \{\mathbf{Ex}^a, \mathbf{Bc}^a\}$, one can define $\mathbf{GNC}^m\mathbf{I}$, $\mathbf{LGNC}^m\mathbf{I}$, $\mathbf{ResGNC}^m\mathbf{I}$ and $\mathbf{BGNC}^m\mathbf{I}$, $\mathbf{LBGNC}^m\mathbf{I}$, $\mathbf{ResBGNC}^m\mathbf{I}$, similar to **NC**-variants.

Note a subtle difference between models $\mathbf{LBGNC}^n$ and $\mathbf{LGNC}^n$: in the model $\mathbf{LBGNC}^n$, the teacher provides the shortest counterexample only if it is smaller than some element of the input, whereas there is no such requirement for $\mathbf{LGNC}^n$ (the same is true also for **NC**-variant).

Note that, similar to Theorems 8 and 10, one can establish corresponding results for the above defined criteria too. For example, for **NCEx**-identification, if $\mathbf{M}$ **NCEx**-identifies L, then there exists a (σ, σ') such that content$(\sigma) \subseteq L$, σ' is a valid sequence of counterexamples for $\mathbf{M}$ on input σ (for the input language being L) and for all (τ, τ') such that $\sigma \subseteq \tau$, $\sigma' \subseteq \tau'$, content$(\tau) \subseteq L$, and τ' is a valid sequence of counterexamples for $\mathbf{M}$ on input τ (for the input language being L), $\mathbf{M}(\tau, \tau') = \mathbf{M}(\sigma, \sigma')$ is a grammar for L. In some cases, we would just refer to σ above as a locking sequence with σ' being implicit.

In the rest of the section, we establish some useful facts about **GNC**-style learners (without requirement for counterexamples being short), as this model is defined here for the first time.

Proposition 16. *Suppose $n \in N$.*

(a) $\mathbf{LNC}^n\mathbf{I} \subseteq \mathbf{LGNC}^n\mathbf{I}$.
(b) $\mathbf{NC}^n\mathbf{I} \subseteq \mathbf{GNC}^n\mathbf{I}$.
(c) $\mathbf{ResNC}^n\mathbf{I} \subseteq \mathbf{ResGNC}^n\mathbf{I}$.
(d) $\mathbf{LSubQ}^n\mathbf{I} \subseteq \mathbf{LGNC}^n\mathbf{I}$.
(e) $\mathbf{SubQ}^n\mathbf{I} \subseteq \mathbf{GNC}^n\mathbf{I}$.
(f) $\mathbf{ResSubQ}^n\mathbf{I} \subseteq \mathbf{ResGNC}^n\mathbf{I}$.

Proof. (a), (b) and (c) follow from the corresponding definitions. As subset queries made by a query learner can be made by a **GNC** learner (by using the query as its conjecture and making the conjecture-subset query), without getting any other counterexamples, (d), (e) and (f) also hold. ∎

Corollary 17. $\mathbf{ResGNC}^1\mathbf{Ex} - \mathbf{LNC}^n\mathbf{Bc}^* \neq \emptyset$.

Proof. Jain and Kinber [10] showed that $\mathbf{ResSubQ}^1\mathbf{Ex} - \mathbf{LNC}^n\mathbf{Bc}^* \neq \emptyset$. Corollary now follows from Proposition 16. ∎

Theorem 18 (*[10]*). *Suppose $n \in N$.*

(a) $\mathbf{ResGNC}^1\mathbf{Ex} - \mathbf{LSubQ}^n\mathbf{Bc}^* \neq \emptyset$.
(b) $\mathbf{ResGNC}^1\mathbf{Bc} - \mathbf{LSubQ}^*\mathbf{Bc}^* \neq \emptyset$.
(c) $\mathbf{ResGNC}^1\mathbf{Ex} - \mathbf{LEquQ}^n\mathbf{Bc}^* \neq \emptyset$.
(d) $\mathbf{ResGNC}^1\mathbf{Ex} - \mathbf{LSupQ}^*\mathbf{Bc}^* \neq \emptyset$.

Proof. Jain and Kinber [10] showed these diagonalizations for $\mathbf{ResNC}^1\mathbf{Ex}$. Theorem now follows using Proposition 16. ∎

(a), (b) above are strongest possible as $\mathbf{ResSubQ}^*\mathbf{Ex}^a = \mathbf{NCEx}^a$ (see [10]), and thus, $\mathbf{ResSubQ}^*\mathbf{Ex}^a$ contains $\mathbf{ResGNCEx}^a$. Similarly, (c) above is strongest as $\mathbf{LEquQ}^*\mathbf{Ex}$ contains $\mathcal{E}$ (see [10]).

Jain and Kinber [10] showed $\mathbf{ResEquQ}^1\mathbf{Ex} \cap \mathbf{ResSupQ}^1\mathbf{Ex} - \mathbf{NCBc} \neq \emptyset$, which also gives us $\mathbf{ResEquQ}^1\mathbf{Ex} \cap \mathbf{ResSupQ}^1\mathbf{Ex} - \mathbf{GNCBc} \neq \emptyset$ and $\mathbf{ResEquQ}^1\mathbf{Ex} \cap \mathbf{ResSupQ}^1\mathbf{Ex} - \mathbf{GNCEx}^* \neq \emptyset$ (note that $\mathbf{LNCEx}^* \subseteq \mathbf{NCBc}$, [11], and **GNC**-model is same as **NC**-model for unbounded number of counterexamples).

Similarly the proof of $\mathbf{ResSupQ}^1\mathbf{Ex} - \mathbf{LNC}^n\mathbf{Bc}^m \neq \emptyset$ in [10] (based on the proof of $(\mathbf{ResEquQ}^1\mathbf{Ex} \cap \mathbf{ResSupQ}^1\mathbf{Ex}) - \mathbf{NCBc} \neq \emptyset$ there), can also be used to show that $(\mathbf{ResSupQ}^1\mathbf{Ex} \cap \mathbf{ResEquQ}^1\mathbf{Ex}) - \mathbf{LGNC}^n\mathbf{Bc}^m \neq \emptyset$. Note that this is the strongest possible result for superset queries, as $\mathbf{ResSupQ}^*\mathbf{Bc}^* = \mathbf{TxtBc}^* \subseteq \mathbf{LGNC}^0\mathbf{Bc}^*$, and $\mathbf{ResEquQ}^n\mathbf{Bc}^* \subseteq \mathbf{ResSubQ}^n\mathbf{Bc}^* \subseteq \mathbf{ResGNC}^n\mathbf{Bc}^*$. (In contrast, note that $\mathbf{ResEquQ}^1\mathbf{Ex} - \mathbf{LNC}^n\mathbf{Bc}^* \neq \emptyset$ [10]).

Theorem 19. *Suppose $n \in N$.* $\mathbf{EquQ}^2\mathbf{Ex} - \mathbf{LGNC}^n\mathbf{Bc}^* \neq \emptyset$.

Proof. Let $\mathcal{L} = \{L \mid (\exists i)[L \subseteq \mathrm{CYL}_i$ and $[L = W_i$ or $(\exists j)[L = W_i \cup \{\langle i, \langle j, x\rangle\rangle \mid x \in N\}]$ or $\mathrm{card}(L) < \infty]]\}$.

It is easy to verify that $\mathcal{L} \in \mathbf{EquQ}^2\mathbf{Ex}$, as a learner can output a grammar for $\emptyset$, until an element in the input appears. If this element is of the form $\langle i, x\rangle$, then the learner asks an equivalence query for W_i. If true, then the learner knows the input language. Otherwise, the learner gets a counterexample $\langle i, \langle j, x\rangle\rangle$ for some j, x. Then the learner asks the query for the language $W_i \cup \{\langle i, \langle j, x\rangle\rangle \mid x \in N\}$. If the answer is yes, then the learner again knows the input language. Otherwise the input language must be finite, and one can easily learn it.

Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{LGNC}^n\mathbf{Bc}^*$-identifies $\mathcal{L}$. Then by Kleene's Recursion Theorem [18] there exists an e such that W_e may be defined as follows. We assume without loss of generality that $\mathbf{M}$ would not ask any more conjecture-subset question after having received n counterexamples, on all inputs, even those outside the class.

Intuitively, the idea of diagonalization is to find an initial segment σ (contained in CYL_e) on which one can force largest number of (least) counterexamples for $\mathbf{M}$. This is achieved by looking for the lexicographically least sequence of counterexample text, where # is taken to be larger than any positive element. During this process, W_e would consist of the initial segments tested for the above ones, but exclude the potential counterexamples. Using the lexicographically least sequence of counterexamples (found in the limit), one can use Proposition 11 along with $W_e \cup \{\langle e, \langle j, x\rangle\rangle \mid x \in N\}$, for an appropriate j, and its finite subsets to do the diagonalization. We now proceed formally.

Let $\sigma_0 = \sigma_0' = \Lambda$. Let W_e^s denote W_e enumerated before stage s. Let $S_0 = \emptyset$. Intuitively, S_s denotes the set of elements we have decided to be out of W_e. Go to stage 0.

Stage s

Invariants we have are

(a) $\mathrm{content}(\sigma_s) \subseteq W_e^s \subseteq \mathrm{CYL}_e - S_s$ and $\mathrm{content}(\sigma_s') \subseteq S_s$.

(b) $|\sigma_s| = |\sigma_s'|$, and σ_s' has at most n non-# entries.

(c) The counterexamples, when present, are correct in the sense that, for $w < |\sigma_s|$, $\sigma_s'(w) \in \{\#\} \cup (W_{\mathbf{M}(\sigma_s[w], \sigma_s'[w])} - W_e)$.

(d) If one treats # > any member of N, then $\sigma_r'\#^\infty$ is lexicographically larger than $\sigma_{r+1}'\#^\infty$. Note that this, along with (b), implies that the number of stages is finite.

1. Dovetail steps 2 and 3 until one of them succeeds. If step 2 succeeds, before step 3 does, if ever, then go to step 4. If step 3 succeeds, before step 2 does, if ever, then go to step 5.
2. Search for a $w < |\sigma_s|$ such that $\mathbf{M}(\sigma_s[w], \sigma_s'[w])$ asks a conjecture-subset question and $W_{\mathbf{M}(\sigma_s[w], \sigma_s'[w])} - W_e^s$ contains an element $z < \sigma_s'(w)$ (where we take # to be ∞).
3. Search for a $\sigma \supseteq \sigma_s$ such that $\mathrm{content}(\sigma) \subseteq \mathrm{CYL}_e - S_s$, and $\mathbf{M}(\sigma, \sigma_s'\#^{|\sigma|-|\sigma_s|})$ asks a conjecture-subset query and $W_{\mathbf{M}(\sigma, \sigma_s'\#^{|\sigma|-|\sigma_s|})}$ enumerates an element z not in $W_e^s \cup \mathrm{content}(\sigma)$.
4. If step 2 succeeds, then let $\sigma_{s+1} = \sigma_s[w]\#$, and $\sigma_{s+1}' = \sigma_s'[w]\diamond z$. Let $S_{s+1} = S_s \cup \{z\}$. $W_e^{s+1} = W_e^s$ and go to stage $s+1$.
5. If step 3 succeeds, then let $\sigma_{s+1} = \sigma\#$, and $\sigma_{s+1}' = \sigma_s'\#^{|\sigma|-|\sigma_s|}z$. Let $S_{s+1} = S_s \cup \{z\}$. Let $W_e^{s+1} = W_e^s \cup \mathrm{content}(\sigma)$ and go to stage $s+1$.

End stage s

It is easy to verify that the invariants are satisfied; especially note that (d) holds as $\sigma_{s+1}'\#^\infty$ is lexicographically smaller than $\sigma_s'\#^\infty$, based on either step 4 or step 5 being executed.

Thus the number of stages is finite. Let s be the last stage that is executed. As step 2 did not succeed, answers given by σ_s' form correct least counterexample sequence for σ_s, for any language L such that $W_e = W_e^s \subseteq L \subseteq \mathrm{CYL}_e - S_s$.

Furthermore, as step 3 did not succeed, for any $\sigma \supseteq \sigma_s$ such that content$(\sigma) \subseteq \mathrm{CYL}_e - S_s$, if $\mathbf{M}(\sigma, \sigma_s'\#^{|\sigma|-|\sigma_s|})$ asks a conjecture-subset question, then $W_{\mathbf{M}(\sigma,\sigma_s'\#^{|\sigma|-|\sigma_s|})} \subseteq W_e^s \cup \mathrm{content}(\sigma)$. It follows that for any text T extending σ_s for a language L such that $W_e^s \subseteq L \subseteq \mathrm{CYL}_e - S_s$, $\sigma_s'\#^\infty$ is a valid sequence of counterexamples. Let j be any number such that S_s does not contain any element of the form $\langle e, \langle j, x\rangle\rangle$. Thus, $\mathbf{M}$ needs to $\mathbf{TxtBc}^*$-identify $W_e \cup \{\langle e, \langle j, x\rangle\rangle \mid x \in N\}$ and all finite subsets of it which contain W_e, without getting any more counterexamples, an impossible task by Proposition 11. ∎

Theorem 20. *Suppose $n \in N$, and* $\mathbf{I} \in \mathbf{Ex}, \mathbf{Bc}$.

(a) $(\mathbf{ResNC}^{n+1}\mathbf{Ex} \cap \mathbf{ResSubQ}^{n+1}\mathbf{Ex} \cap \mathbf{ResEquQ}^{n+1}\mathbf{Ex}) - \mathbf{LGNC}^{n}\mathbf{Bc}^* \neq \emptyset$.
(b) $\mathbf{LGNC}^{n}\mathbf{Ex} - \mathbf{GNC}^{2n-2}\mathbf{Bc}^* \neq \emptyset$.
(c) $\mathbf{GNC}^{n}\mathbf{Ex} - \mathbf{ResGNC}^{2n-2}\mathbf{Bc}^* \neq \emptyset$.
(d) $\mathbf{LGNC}^{n}\mathbf{I} \subseteq \mathbf{ResGNC}^{2n-1}\mathbf{I}$.

Proof. (a) Jain and Kinber [10] showed that $(\mathbf{ResNC}^{n+1}\mathbf{Ex} \cap \mathbf{ResSubQ}^{n+1}\mathbf{Ex} \cap \mathbf{ResEquQ}^{n+1}\mathbf{Ex}) - \mathbf{LNC}^{n}\mathbf{Bc}^* \neq \emptyset$. The proof can be easily modified to show part (a).
(b) Theorem 24 below shows $\mathbf{LBNC}^{n}\mathbf{Ex} - \mathbf{BGNC}^{2n-2}\mathbf{Bc}^m \neq \emptyset$, using a class $\mathcal{C}_n$. $\mathcal{C}_n$ can easily be seen to be in $\mathbf{LGNC}^{n}\mathbf{Ex}$. The diagonalization can be modified to show that $\mathcal{C}_n \notin \mathbf{GNC}^{2n-2}\mathbf{Bc}^*$. Essentially, instead of looking for a counterexample below the largest value in the input, we look for any possible counterexample. Here even diagonalization against $\mathbf{Bc}^*$ works, as $\mathbf{Bc}^*$-identification is enough to guarantee the existence of σ as needed at (steps corresponding to) steps 1.2 and 3.2. We omit the details.
(c) Jain and Kinber [10] showed that $\mathbf{NC}^{n}\mathbf{Ex} - \mathbf{ResNC}^{2n-2}\mathbf{Bc}^* \neq \emptyset$. This proof can be easily modified to show that $\mathbf{GNC}^{n}\mathbf{Ex} - \mathbf{ResGNC}^{2n-2}\mathbf{Bc}^* \neq \emptyset$. We omit the details.
(d) Jain and Kinber [10] showed that $\mathbf{LNC}^{n}\mathbf{I} \subseteq \mathbf{ResNC}^{2n-1}\mathbf{I}$. Similar proof shows this result also. ∎

Thus, below we will deal only with separations/simulations where at least one of the party involves bounded negative counterexamples.

5. Relations among bounded negative counterexample models

In this section we establish relationships between **B**-variants of **NC** and **GNC**-models when any short, or the least short counterexamples, or just the 'no' answers about existence of short counterexamples are used.

First we establish that, similar to the known result about **NC**-model [10], number of counterexamples matters to the extent that $n+1$ 'no' answers used by **BNCEx**-style learners can sometimes do more than n least counterexamples obtained by $\mathbf{LBGNCBc}^*$-style learners.

Theorem 21. *Suppose $n \in N$.* $\mathbf{ResBNC}^{n+1}\mathbf{Ex} - \mathbf{LBGNC}^{n}\mathbf{Bc}^* \neq \emptyset$.

Proof. Proof of $\mathbf{ResNC}^{n+1}\mathbf{Ex} - \mathbf{LNC}^{n}\mathbf{Bc}^* \neq \emptyset$ in [10] can easily be modified to show this result. ∎

The next result gives advantages of **GNC**-model.

Theorem 22. *For all $n, m \in N$,* $\mathbf{ResBGNC}^{1}\mathbf{Ex} - \mathbf{LBNC}^{n}\mathbf{Bc}^m \neq \emptyset$.
$\mathbf{ResBGNC}^{1}\mathbf{Ex} - \mathbf{LBNC}^{n}\mathbf{Ex}^* \neq \emptyset$.

Proof. The proof of $\mathbf{ResSubQ}^{1}\mathbf{Ex} - \mathbf{LNC}^{n}\mathbf{Bc}^*$ from [10] can be easily adopted to prove this theorem (however, only for $\mathbf{Ex}^*$ and $\mathbf{Bc}^m$ cases. The proof for $\mathbf{Bc}^*$ case does not carry over). We omit the details. ∎

Our main results in this section deal with the following problems: if and under which conditions, (a) **B**-learners receiving just 'yes' or 'no' answers can simulate learners receiving (up to) n short (or, possibly, even least short) counterexamples, and (b) learners using arbitrary short counterexamples can simulate the ones receiving (up to) n least short counterexamples. As our results indicate, in both cases such a simulation is quite possible (thus, for example, a "smart" learner can quite compensate for the lack of actual counterexamples) at the expense of just nearly doubling the number of necessary negative answers/counterexamples. More specifically, we establish that, for both tasks (a) and (b), for the $\mathbf{Bc}^m$ and $\mathbf{Ex}^*$-types of learnability, $2n-1$ is the upper and the lower bound on the number of negative answers/examples needed for such a simulation. These results are similar to the corresponding results in [10] for the model **NC**, however, there is also an interesting difference: as it will be shown below, for **Bc***-learnability, the bound $2n-1$ can be lowered to just n (for **NCBc***-learners, the lower bound $2n-1$ still holds).

First we establish the upper bound $2n-1$ for both tasks (a) and (b).

Theorem 23. *For all $n \in N$, $n \geq 1$,*

(a) $\mathbf{LBNC}^n\mathbf{I} \subseteq \mathbf{ResBNC}^{2n-1}\mathbf{I}$.
(b) $\mathbf{LBGNC}^n\mathbf{I} \subseteq \mathbf{ResBGNC}^{2n-1}\mathbf{I}$.

Proof. Proof of $\mathbf{LNC}^n\mathbf{I} \subseteq \mathbf{ResNC}^{2n-1}\mathbf{I}$ from [10] can be used to show this theorem also. ∎

Our next result shows that, for the $\mathbf{Bc}^m$ and $\mathbf{Ex}^*$-types of learnability, the bound $2n-1$ is tight in the strongest sense for the task (b). Namely, we show that **BNC**-learners using n least short counterexamples cannot be simulated by **BGNC**-learners using $2n-2$ (arbitrary short) counterexamples.

Theorem 24. *Suppose $n \in N$ and $n \geq 1$.*

(a) $\mathbf{LBNC}^n\mathbf{Ex} - \mathbf{BGNC}^{2n-2}\mathbf{Bc}^m \neq \emptyset$.
(b) $\mathbf{LBNC}^n\mathbf{Ex} - \mathbf{BGNC}^{2n-2}\mathbf{Ex}^* \neq \emptyset$.

Proof. This proof is a modification of the proof of $\mathbf{LNC}^n\mathbf{Ex} - \mathbf{NC}^{2n-2}\mathbf{Bc}^* \neq \emptyset$ from [10]. We give details as there are some subtilties, and also the result does not carry over for $\mathbf{Bc}^*$. Without loss of generality assume that the pairing function is increasing in all its arguments.

Let
$E = \{\langle n, x, y\rangle \mid x, y \in N\}$.
$L_{i,k} = \{\langle i, k, x\rangle \mid x \in N\}$.
$X_i = L_{i,0}$.
$Y_i^j = \{\langle i, 0, x\rangle \mid x < 3j\} \cup L_{i,j+1}$.
$Z_i^{j,k} = \{\langle i, 0, x\rangle \mid x < 3j+1\} \cup \{\langle i, j+1, x\rangle \mid x \leq k\}$.
$U_i^j = \{\langle i, 0, x\rangle \mid x < 3j+2\}$.
$\mathcal{L}_i = \{X_i\} \cup \{Y_i^j \mid j \in N\} \cup \{U_i^j \mid j \in N\} \cup \{Z_i^{j,k} \mid j, k \in N\}$.
$\mathcal{C}_n = \{L \mid [L$ is formed by picking one language from each $\mathcal{L}_i$, $i < n$, and then taking the union of these languages along with $E]\}$.

Here note that U_i^j are growing initial segments of X_i, and $\bigcup_j U_i^j = X_i$. Thus, by Proposition 11, for any learner it is impossible to $\mathbf{TxtBc}^m$ ($\mathbf{TxtEx}^*$)-identify X_i as well as all of U_i^j, from just positive data.

Also $Z_i^{j,k} - \{\langle i, 0, 3j\rangle\}$ are growing initial segments of Y_i^j.

Intuitively, each $L \in \mathcal{L}_i$ is either X_i or contains an initial segment of X_i, and the least element from $X_i - L$ indicates the form of L (i.e., whether it is Y_i^j, $Z_i^{j,k}$ or U_i^j, for some j, k). This allows for easy learnability when one gets n least counterexamples. However, it will be shown below that $(2n-2)$ negative answers are not enough for learning the above class. E has been added to the languages just to ensure that the language is infinite, and thus negative counterexamples from X_i, if present, can eventually be obtained (because they become smaller than the largest element of the input at some point).

A learner can $\mathbf{LBNC}^n\mathbf{Ex}$-identify the class $\mathcal{C}_n$ as follows. On input (σ, σ'), do as follows.

Let $A' = \{i \mid (\exists j)[\langle i, 0, 3j\rangle \in \mathrm{content}(\sigma')]\}$. Let $A'' = \{i \mid (\exists j)[\langle i, 0, 3j+1\rangle \in \mathrm{content}(\sigma')$ or $\langle i, 0, 3j+2\rangle \in \mathrm{content}(\sigma')]\}$.

It would be the case that for input from $\mathcal{C}_n$ the sets A', A'' are disjoint subsets of $\{i \mid i < n\}$ (see below). For $i \in A'$, let j_i be such that $\langle i, 0, 3j_i\rangle \in \mathrm{content}(\sigma')$.

Output a (standard) grammar for the language:

$$E \cup \bigcup_{i \in \{r \mid r < n\} - A' - A''} X_i \cup \bigcup_{i \in A'} Y_i^{j_i} \cup \bigcup_{i \in A''} [\mathrm{content}(\sigma) \cap \{\langle i, x, y\rangle \mid x, y \in N\}].$$

Now consider any input language $L \in \mathcal{C}_n$. By induction on the length of the input, we claim that counterexamples received would only be of the form $\langle i, 0, z\rangle$, where $i < n$. Furthermore, for any given i, there is at most one such counterexample of the form $\langle i, 0, z\rangle$ that the learner will receive – ensuring that A', A'' are disjoint as claimed earlier.

Now, consider any $i < n$. We consider the following cases.

Case 1: There is no counterexample ever received from X_i.

In this case the language from $\mathcal{L}_i$, which is a subset of L, must be X_i. Furthermore, for any future input, we will never have a counterexample of the form $\langle i, x, y\rangle$, and thus i will never be placed in A', A''. Thus, X_i would be contained in the conjectured language.

Case 2: There is a counterexample of the form $\langle i, 0, 3j\rangle$.

In this case the language from $\mathcal{L}_i$ which is a subset of L must be Y_i^j. Also, i will be placed in A'. Furthermore, we will never have a counterexample of the form $\langle i, x, y\rangle$, for any future input. Thus, Y_i^j would be contained in the conjectured language.

Case 3: There is a counterexample of the form $\langle i, 0, 3j + 1\rangle$ or $\langle i, 0, 3j + 2\rangle$.

In this case the language from $\mathcal{L}_i$, which is a subset of L, must be finite. Also, i will be placed in A''. Furthermore, we will never have a counterexample of the form $\langle i, x, y\rangle$, for any future input, due to the form of conjectures made by the learner.

From the above cases, it is easy to verify that induction hypothesis would be satisfied, and eventually the learner would converge to a grammar for L. Thus, $\mathcal{C}_n \in \mathbf{LBNC}^n\mathbf{Ex}$.

We now show that $\mathcal{C}_n \notin \mathbf{BGNC}^{2n-2}\mathbf{Bc}^m$ or $\mathbf{BGNC}^{2n-2}\mathbf{Ex}^*$. So suppose by way of contradiction $\mathbf{M}$ $\mathbf{BGNC}^{2n-2}\mathbf{Bc}^m$-identifies ($\mathbf{BGNC}^{2n-2}\mathbf{Ex}^*$-identifies) $\mathcal{L}$.

Intuitively, the idea of the proof is that we try to force two counterexamples in choosing a member of each $\mathcal{L}_i$, $i < n - 1$, which forms the part of the diagonalizing language. The choice of a member of $\mathcal{L}_{n-1}$ can then force non-learnability using Proposition 11. To force the above mentioned two counterexamples for the selected part from each $\mathcal{L}_i$, note that, by Proposition 11, the learner $\mathbf{M}$ needs to ask a conjecture-subset question for a language containing an infinite subset of X_i to be able to learn X_i as well as U_i^j, which may form a part of the diagonalizing language. This allows us to force one counterexample – while committing the language part from $\mathcal{L}_i$ to be one of Y_i^j or $Z_i^{j,k}$, for a fixed j. We then use a similar trick, but for $X_{n-1}/U_{n-1}^{j'}$, to force the learner to output an hypothesis which contains an infinite part of Y_i^j, and ask a conjecture-subset question. This would allow us to get the second counterexample (which is from Y_i^j, and thus does not constrain the choice from $\mathcal{L}_n$). Details of the implementation of the above idea are actually more complex as one needs to be careful about the bound on the counterexamples, as well as consider the possibilities of other counterexamples being there. We now proceed formally.

Let I_m denote the set $\{x \mid x \leq m\}$. We will construct the diagonalizing language L in stages.

In stage $s < n - 1$, we will try to fix $F_s \in \mathcal{L}_s$, which is contained in the diagonalizing language L. Initially, let $\sigma_0 = \Lambda$, $\sigma_0' = \Lambda$. σ_s denote the initial segment of a target diagonalizing language constructed before the beginning of stage s. σ_s' would denote the sequence of counterexamples/# provided to $\mathbf{M}$ on input σ_s. The invariants below provide some properties of these sets, in particular (d) states the set of languages which are still possible to use for diagonalization. Intuitively, (d) states that for $r \geq s$, one can still choose all possible members of $\mathcal{L}_r$, except some which are ruled out due to counterexamples or positive data already in σ_s, σ_s'.

For $s < n - 1$, inductively define F_s and $\sigma_{s+1}, \sigma_{s+1}'$ as follows.
(* The construction is non-effective. *)
(* Following invariants will be satisfied:
(a) For $r < s$, $F_r \in \mathcal{L}_r$.
(b) content$(\sigma_s) \subseteq E \cup \bigcup_{r<s} F_r \cup \bigcup_{s\leq r<n} X_r$.
(c) σ_s' contains at most $(|\sigma_s'| - 2s)$ #s. Thus, at least $2s$ counterexamples have already been provided.

(d) For $s \leq r < n$, for all possibilities for $H_r \in \mathcal{L}_r$ such that $H_r \notin \{Y_r^j, Z_r^{j,k} \mid k \in N$ and $[\langle r, 0, 3j\rangle \leq \max(\text{content}(\sigma_s))$ or $(\exists x)[\langle r, j+1, x\rangle \in \text{content}(\sigma_s')]]\}$, answers given to $\mathbf{M}$ via σ_s', on input σ_s are consistent with the input language being $E \cup \bigcup_{r<s} F_r \cup \bigcup_{s\leq r<n} H_r$.

*)

1. If there exists a $\sigma \supseteq \sigma_s$ such that $\text{content}(\sigma) \subseteq E \cup \bigcup_{r<s} F_r \cup \bigcup_{s\leq r<n} X_r$, $\mathbf{M}(\sigma, \sigma_s' \#^{|\sigma|-|\sigma_s|})$ asks a conjecture-subset question, and
   $W_{\mathbf{M}(\sigma,\sigma_s'\#^{|\sigma|-|\sigma_s|})} - I_{\max(\text{content}(\sigma))}$ contains an element of the form $\langle i', j', k'\rangle$ such that one of the following conditions is satisfied:
   1.1. $\langle i', j', k'\rangle \notin (E \cup \bigcup_{r<s} F_r \cup \bigcup_{s\leq r<n} X_r)$
   1.2. Not 1.1, and for some $j \in N$,
   (i) $i' = s$, $j' = 0$, $k' \geq 3j+3$,
   (ii) for all x, $\langle s, j+1, x\rangle \notin \text{content}(\sigma_s')$,
   (iii) $\max(\text{content}(\sigma) - \{\max(\text{content}(\sigma))\}) < \langle s, 0, 3j\rangle$, and
   (iv) $\max(\text{content}(\sigma)) \notin X_s$, and
   (v) $\mathbf{M}(\sigma[w], \sigma_s'\#^{w-|\sigma_s|})$ does not ask a conjecture-subset question for $\min(\{t \mid \sigma(t) = \max(\text{content}(\sigma))\}) < w < |\sigma|$ (that is, since the maximal element in content(σ) was seen, σ is the first point at which $\mathbf{M}$ asks a subset-conjecture question).
2. Then, pick shortest such σ (we will argue below that there must exist such a σ).
   Let $\tau = \sigma\#$ and $\tau' = \sigma_s'\#^{|\sigma|-|\sigma_s|}\langle i', j', k'\rangle$.
   If step 1.1 succeeded then let j be such that (i) $\langle s, j+1, x\rangle \notin \text{content}(\sigma_s')$ for all x, and (ii) $\max(\text{content}(\sigma)) < \langle s, 0, 3j\rangle$.
   (* Note that answers given by τ' are consistent with invariant (d) for $F_s = Y_s^j$, or $F_s = Z_s^{j,k}$ for any k, as steps 1.1 and 1.2 did not succeed on any proper prefix of σ. *)
3. If there exists a $\sigma \supseteq \tau$ such that $\text{content}(\sigma) \subseteq E \cup Y_s^j \cup \bigcup_{r<s} F_r \cup \bigcup_{s<r<n} X_r$, and $\mathbf{M}(\sigma, \tau'\#^{|\sigma|-|\tau|})$ asks a conjecture-subset question, and
   $W_{\mathbf{M}(\sigma,\tau'\#^{|\sigma|-|\tau|})} - I_{\max(\text{content}(\sigma))}$ contains an element of the form $\langle i'', j'', k''\rangle$ such that one of the following conditions is satisfied:
   3.1. $\langle i'', j'', k''\rangle \notin E \cup Y_s^j \cup \bigcup_{r<s} F_r \cup \bigcup_{s<r<n} X_r$,
   3.2. Not 3.1 and $i'' = s$, $j'' = j+1$, and $k'' > \max(\{k \mid \langle s, j+1, k\rangle \in \text{content}(\sigma)\})$,
4. Then, pick a shortest such σ (we will argue below that there must exist such a σ).
   Let $\sigma_{s+1} = \sigma\#$ and $\sigma_{s+1}' = \tau'\#^{|\sigma|-|\tau|}\langle i'', j'', k''\rangle$.
   If 3.1 holds, then let $F_s = Y_s^j$.
   Else (i.e., 3.2 holds), then let $F_s = Z_s^{j,k}$, where $k = \max(\{x \mid \langle s, j+1, x\rangle \in \text{content}(\sigma)\})$.
   (* Note that we give counterexample $\langle i'', j'', k''\rangle$ to $W_{\mathbf{M}(\sigma,\tau'\#^{|\sigma|-|\tau|})}$. *)
   (* Note that answers given by σ_{s+1}' are consistent with invariant (d) for each of above choices of F_s, as step 3.1 and 3.2 did not succeed on any proper prefix of σ. *)

End

It is easy to verify that the invariants are maintained by the construction. Specially note that the invariant (d) is maintained as explained by comments in the construction above.

We first claim that the above construction finishes for every $s < n-1$ (i.e., σ_{n-1}, σ_{n-1}' get defined). If not, then let s be the least number such that stage s starts but does not finish.

Suppose the 'If' statement at step 2 does not hold. Now $\mathbf{M}$ must $\mathbf{BGNC}^{2n-2}\mathbf{Bc}^m$-identify ($\mathbf{BGNC}^{2n-2}\mathbf{Ex}^*$-identify) the language $L = E \cup \bigcup_{r<s} F_r \cup \bigcup_{s\leq r<n} X_r$, which is a member of $\mathcal{C}_n$. Suppose γ, extending σ_s, is a $\mathbf{BGNC}^{2n-2}\mathbf{Bc}^m$-locking sequence ($\mathbf{BGNC}^{2n-2}\mathbf{Ex}^*$-locking sequence) for $\mathbf{M}$ on L, where the answers provided beyond σ_s' are always # (i.e., yes whenever the conjecture-subset question is asked). Without loss of generality assume that $W_{\mathbf{M}(\gamma,\sigma_s'\#^{|\gamma|-|\sigma_s'|})}$ contains $L - I_{\max(\gamma)}$, except for maybe m elements (this clearly holds for $\mathbf{BGNC}^{2n-2}\mathbf{Bc}^m$-identification; for $\mathbf{BGNC}^{2n-2}\mathbf{Ex}^*$-identification, one can just take an appropriate extension of γ to ensure this – the maximal element in the extension is larger than the maximal element of L that is missing from $W_{\mathbf{M}(\gamma,\sigma_s'\#^{|\gamma|-|\sigma_s'|})}$). Let j be such that $3j+2 > \max(\{x \mid \langle s, 0, x\rangle \in \text{content}(\gamma)\})$. Let H be an increasing text for $U_s^j \cup E \cup \bigcup_{r<s} F_r \cup \bigcup_{s<r<n} X_r$. Let G be a text for a subset of E such that $G(w) > \max(\{\langle s, 0, 3j' + 3j + 3 + m + 1\rangle\} \cup \text{content}(\gamma))$, where $3j' > \max(\{t \mid \langle s, 0, t\rangle \in \text{content}(H[w+1])\})$. Let $T = \gamma G(0)H(0)G(1)H(1)\ldots$, and $T' = \sigma_s'\#^\infty$. If $\mathbf{M}(T[w], T'[w])$ does not ask a conjecture-subset question for $w \geq |\gamma|$, then $\mathbf{M}$ does not $\mathbf{BGNC}^{2n-2}\mathbf{Bc}^m$ ($\mathbf{BGNC}^{2n-2}\mathbf{Ex}^*$)-identify $U_s^j \cup E \cup \bigcup_{r<s} F_r \cup \bigcup_{s<r<n} X_r$, as the counterexamples provided by T' are valid for input language being $U_s^j \cup E \cup \bigcup_{r<s} F_r \cup \bigcup_{s<r<n} X_r$, but $\mathbf{M}$ (beyond input γ) outputs only grammars for finite variants of $E \cup \bigcup_{r<s} F_r \cup \bigcup_{s\leq r<n} X_r$. On the other hand, if $\mathbf{M}$ does ask a conjecture-subset question at $(T[|\gamma|+1+2w+v], T'[|\gamma|+1+2w+v])$, where $2w+v$ is minimal such number with $w \in N$ and $v \in \{0,1\}$, then $T[|\gamma|+1+2w+v]$ qualifies as being σ in step 1.2. (To see this note that, for some j', $G(w) > \langle s, 0, 3j'+3j+3+m+1\rangle$, where $3j' > \max(\{t \mid \langle s,0,t\rangle \in \text{content}(H[w+1])\})$, and $W_{\mathbf{M}(T[|\gamma|+1+2w+v], T'[|\gamma|+1+2w+v])}$ misses out at most m of $\{\langle s, 0, 3j'+3+x\rangle \mid x \leq m+1\}$ due to the locking sequence property of γ on L).

So assume that step 2.1 or 2.2 did succeed. Suppose the 'If' statement at step 3 does not hold. Now $\mathbf{M}$ must $\mathbf{BGNC}^{2n-2}\mathbf{Bc}^m$-identify ($\mathbf{BGNC}^{2n-2}\mathbf{Ex}^*$-identify) the language $L = Y_s^j \cup E \cup \bigcup_{r<s} F_r \cup \bigcup_{s<r<n} X_r$, which is a member of $\mathcal{C}_n$. Suppose γ, extending τ, is a $\mathbf{BGNC}^{2n-2}\mathbf{Bc}^m$-locking sequence ($\mathbf{BGNC}^{2n-2}\mathbf{Ex}^*$-locking sequence) for $\mathbf{M}$ on L, where the answers provided beyond τ' are always # (i.e., yes whenever the conjecture-subset question is asked). Without loss of generality assume that $W_{\mathbf{M}(\gamma, \tau'\#^{|\gamma|-|\tau'|})}$ contains $L - I_{\max(\gamma)}$, except for maybe m elements (this clearly holds for $\mathbf{BGNC}^{2n-2}\mathbf{Bc}^m$-identification; for $\mathbf{BGNC}^{2n-2}\mathbf{Ex}^*$-identification, one can just take an appropriate extension of γ to ensure this).

Let j' be such that $\langle n-1, 0, j'\rangle > \max(\text{content}(\gamma))$, and $\langle s, j+1, j'\rangle > \max(\text{content}(\gamma))$. Let H be an increasing text for $U_{n-1}^{j'} \cup Y_s^j \cup E \cup \bigcup_{1\leq r<s} F_s \cup \bigcup_{s<r<n-1} X_r$. Let G be a text for a subset of E such that $G(w) > \max(\{\langle s, j+1, 3j''+3j'+3+m+1\rangle\} \cup \text{content}(\gamma))$, where $3j'' > \max(\{t \mid \langle s, j+1, t\rangle \in \text{content}(H[w+1])\})$. Let $T = \gamma G(0)H(0)G(1)H(1)\ldots$, and $T' = \sigma_s'\#^\infty$. If $\mathbf{M}(T[w], T'[w])$ does not ask a conjecture-subset question for $w \geq |\gamma|$, then $\mathbf{M}$ does not $\mathbf{BGNC}^{2n-2}\mathbf{Bc}^m$ ($\mathbf{BGNC}^{2n-2}\mathbf{Ex}^*$)-identify $U_{n-1}^{j'} \cup Y_s^j \cup E \cup \bigcup_{r<s} F_r \cup \bigcup_{s<r<n-1} X_r$, as the counterexamples provided by T' are valid for input language being $U_{n-1}^{j'} \cup Y_s^j \cup E \cup \bigcup_{r<s} F_r \cup \bigcup_{s<r<n-1} X_r$, but $\mathbf{M}$ (beyond input γ) outputs only grammars for finite variants of $Y_s^j \cup E \cup \bigcup_{r<s} F_r \cup \bigcup_{s<r<n} X_r$. On the other hand, if $\mathbf{M}$ does ask a conjecture-subset question at $(T[|\gamma|+1+2w+v], T'[|\gamma|+1+2w+v])$, where $2w+v$ is minimal such number with $w \in N$ and $v \in \{0,1\}$, then $T[|\gamma|+1+2w+v]$ qualifies as being σ in step 3.2. (To see this note that, for some j'', $G(w) > \langle s, j+1, 3j''+3j'+3+m+1\rangle$, where $3j'' > \max(\{t \mid \langle s, j+1, t\rangle \in \text{content}(H[w+1])\})$, and $W_{\mathbf{M}(T[|\gamma|+1+2w+v], T'[|\gamma|+1+2w+v])}$ misses out at most m of $\{\langle s, j+1, 3j''+3+x\rangle \mid x \leq m+1\}$ due to the locking sequence property of γ on L).

Thus, $\sigma_{n-1}, \sigma_{n-1}'$ must get defined. Now, on the input $(\sigma_{n-1}, \sigma_{n-1}')$, $\mathbf{M}$ has already received $2n-2$ negative counterexamples (two counterexamples each during the definition of σ_{s+1}, for $s < n-1$). Now, $\mathbf{M}$ needs to $\mathbf{BGNC}^{2n-2}\mathbf{Bc}^*$-identify $F_{n-1} \cup E \cup \bigcup_{r<n} F_r$, for every possible $F_{n-1} \in \mathcal{L}_{n-1}$, without receiving any more counterexamples. This is impossible, as by Proposition 11, no machine can $\mathbf{TxtBc}^*$-identify $X_{n-1} \cup E \cup \bigcup_{r<n-1} F_r$, and $U_{n-1}^j \cup E \cup \bigcup_{r<n-1} F_r$, for all j. ∎

Now we show that the bound $2n-1$ on the number of negative answers is tight for $\mathbf{Bc}$ and $\mathbf{Ex}^*$-types of learnability when $\mathbf{ResBNC}$-learners try to simulate $\mathbf{BNC}^n$-learners. In fact, we show this in the strongest possible way: there are $\mathbf{BNC}^n\mathbf{Ex}$-learners that cannot be simulated by $\mathbf{ResBNC}^{2n-2}\mathbf{Bc}^m$ or $\mathbf{ResBNC}^{2n-2}\mathbf{Ex}^*$-learners (our next theorem does it just for $\mathbf{Bc}$ rather than for $\mathbf{Bc}^m$; the case of $\mathbf{Bc}^m$ is addressed in Theorem 26).

Theorem 25. *Suppose $n \in N$.* $\mathbf{BNC}^n\mathbf{Ex} - (\mathbf{ResBNC}^{2n-2}\mathbf{Bc} \cup \mathbf{ResBNC}^{2n-2}\mathbf{Ex}^*) \neq \emptyset$.

Proof. We assume without loss of generality that pairing function is increasing in all its arguments. Recall that $\langle x, y, z\rangle = \langle x, \langle y, z\rangle\rangle$. Thus, $\text{CYL}_j = \{\langle j, x, y\rangle \mid x, y \in N\}$, and $\langle\cdot,\cdot,\cdot\rangle$ is increasing in all its arguments.

Consider $\mathcal{L}$ defined as follows.

For each $L \in \mathcal{L}$, there exists a set S, $\text{card}(S) \leq n$, such that the conditions (1)–(3) hold.

(1) $L \subseteq \bigcup_{j\in S} \text{CYL}_j$.
(2) $L \cap \text{CYL}_j \cap \{\langle j, 0, x\rangle \mid x \in N\}$ contains exactly one element for each $j \in S$. Let this element be $\langle j, 0, \langle p_j, q_j\rangle\rangle$.
(3) For each $j \in S$
  (3A) W_{p_j} is a grammar for $L \cap \text{CYL}_j$ or

(3B) $W_{p_j} \not\subseteq L$ and $W_{p_j} - L$ consists only of elements of the form $\langle j, 1, 2x\rangle$ or only of elements of the form $\langle j, 1, 2x+1\rangle$. Furthermore at least one such element is smaller than $\max(L)$ (where $\max(L)$ is taken to be ∞, for infinite L). If this element is of the form $\langle j, 1, 2z\rangle$, then $W_{q_j} = L \cap \mathrm{CYL}_j$. Otherwise, $L \cap \mathrm{CYL}_j$ is finite.

Intuitively, L may be considered as being divided into up to n parts, each part being subset of a cylinder, where each part satisfies the properties as given in (2) and (3).

Above class of languages can be seen to be in $\mathbf{BNC}^n\mathbf{Ex}$ as follows. On input σ, for each j such that content(σ) contains an element of CYL_j, find p_j and q_j as defined in condition 2 above (if σ does not contain any element of the form $\langle j, 0, \langle p_j, q_j\rangle\rangle$, then grammar for $\emptyset$ is output on σ). Then for each of these j, learner computes a grammar for:

(a) W_{p_j} (if it has not received any counterexample from CYL_j),
(b) W_{q_j} (if the negative counterexample from CYL_j is of the form $\langle j, 1, 2z\rangle$), and
(c) content$(\sigma) \cap \mathrm{CYL}_j$, otherwise.

Then, the learner outputs a grammar for the union of the languages enumerated by the grammars computed for each j above. It is easy to verify that the above learner gets at most one counterexample from each CYL_j such that CYL_j intersects with the input language, and thus $\mathbf{BNC}^n\mathbf{Ex}$-identifies $\mathcal{L}$.

We now show that $\mathcal{L} \notin \mathbf{ResBNC}^{2n-2}\mathbf{Bc} \cup \mathbf{ResBNC}^{2n-2}\mathbf{Ex}^*$. Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{ResBNC}^{2n-2}\mathbf{Bc}$ ($\mathbf{ResBNC}^{2n-2}\mathbf{Ex}^*$)-identifies $\mathcal{L}$.

Intuitively, for each $j \in S$ (except the last one placed in S) we would try to force two counterexamples for $\mathbf{M}$ on the diagonalizing language. For forcing the two counterexamples with respect to $j \in S$, the following is done. First, two sets, W_{p_j} and W_{q_j} are being constructed. Now, suppose τ_i denotes the part of input text already constructed (for the previous i elements placed into S). If one ever finds that $\mathbf{M}$ on some appropriate extension σ of τ_i has output a conjecture enumerating an element outside W_{p_j} plus content(σ), then one can force two counterexamples by freezing such σ (which will lead to a counterexample) and then making W_{p_j} enumerate all elements of the form $\langle j, 1, 2x+1\rangle$ (except those elements which have been used for counterexamples) and then using Proposition 11 along with condition (3B) above, which will force one more counterexample (see step 2 and Case 1 below). If $\mathbf{M}$ never outputs a conjecture which enumerates an element outside W_{p_j} and the input data seen so far, then at each stage s in step 3 below, we try to find two extensions (α_s and γ_s at stage s) on which the learner outputs a conjecture containing an element of the form $\langle j, 1, 2x\rangle$ which does not belong to the input data seen so far (but is bounded by the largest element in the input). These elements are respectively, $\langle j, 1, y_s\rangle$ and z_s in the construction below. To achieve this, we first make W_{p_j} to contain more and more elements of the form $\langle j, 1, 2x\rangle$ (which would allow us to get α_s and $\langle j, 1, y_s\rangle$ above), and then make W_{q_j} to contain more and more elements of the form $\langle j, 1, 2x\rangle$ (which would allow us to get γ_s and z_s above), assuming that the learner learns the above parts W_{p_j} and W_{q_j} respectively (where W_{q_j} would not contain $\langle j, 1, y_s\rangle$) – see steps 3.1 and 3.2 below. If $z_s \notin W_{q_j}$ (in particular $z_s = \langle j, 1, y_s\rangle$), then one can take W_{q_j} to be the diagonalizing part, as this would have forced two counterexamples (see step 3.3, If part and Subcase 3.2 below). If $z_s \in W_{q_j}$ (only tested via $z_s \neq \langle j, 1, y_s\rangle$ in construction below), then note that we may not have yet achieved two counterexamples, as the element $\langle j, 1, y_s\rangle$ which is in W_{p_j} but missing from the input, has y_s as even (see condition (3B) above). To circumvent this problem, we place an element $\langle j, 1, r_s\rangle$ (which is different from z_s) in W_{p_j}. Here r_s would be odd ($r_s = y_s + 1$ or $y_s + 3$; we ensure that $\langle j, 1, r_s\rangle$ would still be below the maximal element in α_s). If $\langle j, 1, r_s\rangle$ is ever enumerated by the conjecture of $\mathbf{M}$ at α_s, then we have achieved two counterexamples – one at α_s (where counterexample taken is $\langle j, 1, r_s\rangle$, which would not be in the diagonalizing language), and one at γ_s – the diagonalizing language would be the finite set content$(\gamma_s) \cup W_{p_j} - \{\langle j, 1, r_s\rangle\}$ (see step 3.4 and Subcase 3.1 below). If the conjecture of $\mathbf{M}$ at α_s never outputs $\langle j, 1, r_s\rangle$, then we continue with the next stage trying a similar process again. If we have infinitely many stages, then one can get the diagonalizing language as W_{p_j}, since conjectures of $\mathbf{M}$ at α_s miss out the element $\langle j, 1, r_s\rangle$ (see Case 2 below). Here note that checking of whether the conjecture of $\mathbf{M}$ at α_s enumerates $\langle j, 1, r_s\rangle$ cannot be done effectively. Thus, in each stage, step 3.4 below checks if some earlier $\langle j, 1, y_t\rangle$ has been enumerated by the conjecture at α_t. We now proceed formally.

Let $I_m = \{x \mid x \leq m\}$.

In the construction below in the definition of τ_i, we will give the exact counterexample to $\mathbf{M}$. This is for ease of presentation (and only gives extra power to $\mathbf{M}$). (However, while exploring the different possibilities, for τ_{i+1}, we will not give the exact value of negative counterexample; these counterexamples only get finalized when τ_{i+1} actually gets defined.)

Initially let $\tau_0 = \tau'_0 = \Lambda$. We will aim to inductively define τ_{i+1}, τ'_{i+1} for $i = 0$ to $i = n-2$ below. Intuitively, τ'_i denotes the negative counterexamples received by $\mathbf{M}$ on conjectures made on input τ_i.

τ_i, τ'_i (if defined) will satisfy the following properties.

(A) $|\tau_i| = |\tau'_i|$ and $\tau_i \subseteq \tau_{i+1}$ and $\tau'_i \subseteq \tau'_{i+1}$ (if defined).

(B) $\text{content}(\tau_i) \cap \text{content}(\tau'_i) = \emptyset$.

(C) τ'_i contains at least $2i$ answers 'no'.

(D) Let $S_i = \{j \mid (\text{content}(\tau_i) \cup \text{content}(\tau'_i)) \cap \text{CYL}_j \neq \emptyset\}$. Then answers received by $\mathbf{M}$ (as given by τ'_i) are consistent with any language L such that $\text{content}(\tau_i) \subseteq L \subseteq \text{content}(\tau_i) \cup \bigcup_{j \notin S_i} \text{CYL}_j$. (That is, for each such input L, for each $\sigma \subseteq \tau_i$, either $\tau'_i(|\sigma|)$ is an element from $W_{\mathbf{M}(\sigma,\tau'_i[|\sigma|])} \cap I_{\max(\text{content}(\sigma))} - \text{content}(\tau_i)$ or $W_{\mathbf{M}(\sigma,\tau'_i[|\sigma|])}$ does not contain any element from $I_{\max(\text{content}(\sigma))} - \text{content}(\tau_i)$).

(E) When defined, $\text{content}(\tau_{i+1}) - \text{content}(\tau_i)$ is a subset of some CYL_j and forms $L \cap \text{CYL}_j$ for the diagonalizing language L (and thus this part satisfies (2) (giving p_j, q_j) and ((3A) or (3B)) above (in fact it satisfies (3B))).

For $i \le n-2$, we will inductively define τ_{i+1} (and τ'_{i+1}), non-effectively, based on a case analysis below.

So suppose τ_i has been defined. Pick a $j \notin S_i$ such that $\langle j, 0, 0\rangle > \max(\text{content}(\tau_i))$. By implicit use of Kleene's Recursion Theorem [18] choose a p_j, q_j such that W_{p_j}, W_{q_j} can be defined as follows.

Definition of W_{p_j}, W_{q_j}

1. Enumerate $\langle j, 0, \langle p_j, q_j\rangle\rangle$ into W_{p_j}.
   Dovetail steps 2 and 3, until step 2 exits. Here we assume that each individual iteration in for loop of step 2 (the portion between 'Atomic and End Atomic') is atomic, and executed at one go without intermediate execution of step 3 (note that this is fine, as each iteration of the for loop is finite).
2. For $t = 1$ to ∞ Do:
   Atomic:
   If there exists a $\sigma \supseteq \tau_i$, such that
   (a) $\max(\text{content}(\sigma)) \le t$,
   (b) $|\sigma| \le t$,
   (c) $\text{content}(\sigma) \subseteq (\text{content}(\tau_i) \cup W_{p_j}$ enumerated up to now), and
   (d) $W_{\mathbf{M}(\sigma,\tau'_i\#^{|\sigma|-|\tau_i|})}$ enumerates, within t steps, an element in $I_{\max(\text{content}(\sigma))} - (\text{content}(\tau_i) \cup W_{p_j}$ enumerated up to now).
   Then pick least such σ, stop dovetailing step 3 and proceed to step 2.1
   End Atomic
   End For
   2.1. Enumerate $\{\langle j, 1, 2x+1\rangle \mid x \in N\} - I_{\max(\text{content}(\sigma))}$ into W_{p_j} and exit the construction.
3. Let $W^s_{p_j}$ and $W^s_{q_j}$ denote W_{p_j} and W_{q_j} enumerated before stage s.
   We will also (try to) define $\sigma_s, \alpha_s, \gamma_s, z_s, r_s, y_s$ in each of the stages below.
   Invariants (when the corresponding values are defined):
   (F) $\text{content}(\tau_i) \cup W^s_{p_j} \subseteq \text{content}(\sigma_s) \subseteq \text{content}(\tau_i) \cup W_{p_j}$.
   (G) $\tau_i \subseteq \sigma_s \subseteq \alpha_s \subseteq \sigma_{s+1}$.
   (H) $\alpha_s \subseteq \gamma_s$.
   (I) y_s is always even, and $r_s \in \{y_s + 1, y_s + 3\}$.
   (J) W_{p_j} will not contain any of z_s's, except maybe z_t for the last stage t which is executed (see step 3.3).
   (K) $W_{\mathbf{M}(\alpha_s,\tau'_i\#^{|\alpha_s|-|\tau_i|})}$ enumerates $\langle j, 1, y_s\rangle$, which is in $I_{\max(\text{content}(\alpha_s))} - \text{content}(\alpha_s)$.
   (L) $\max(\text{content}(\alpha_s[|\alpha_s|-1])) < \langle j, 1, y_s\rangle < \langle j, 1, r_s\rangle < \max(\text{content}(\alpha_s))$ and $\langle j, 1, r_s\rangle$ belongs to $W^{s+1}_{p_j}$.
   (M) $\langle j, 1, r_s\rangle$ and $\langle j, 1, y_s\rangle$ are not in $\text{content}(\gamma_s)$.
   (N) $W_{\mathbf{M}(\gamma_s,\tau'_i\#^{|\alpha_s|-|\tau_i|}\text{'no'}\#^{|\gamma_s|-|\alpha_s|-1})}$ enumerates $z_s \in I_{\max(\gamma_s)} - \text{content}(\gamma_s)$.
   (* 'no' above is the no answer (without explicitly stating the value of counterexample). *)
   Go to stage 0.
   Begin Stage s
   3.1 If $s = 0$, then let σ_0 be an extension of τ_i such that $\text{content}(\sigma_0) = \text{content}(\tau_i) \cup W^0_{p_j}$. Otherwise, let σ_s be a proper extension of α_{s-1} such that $\text{content}(\sigma_s) = W^s_{p_j} \cup \text{content}(\alpha_{s-1})$.

Let $x_s = 1 + \max(\{x \mid \langle j, 1, x\rangle \in W^s_{p_j} \cup W^s_{q_j}\} \cup \{z_t \mid t < s\} \cup \text{content}(\sigma_s))$.
(* Intuitively, x_s is large enough so that the construction below does not interfere with earlier enumerations. *)
Enumerate more and more of $\langle j, 1, 2x\rangle$ such that $2x > x_s$ into W_{p_j}, until a $\alpha_s \supseteq \sigma_s$, and an even $y_s > x_s$ are found such that:
(* Note that $2x > x_s$ ensures that $\langle j, 1, 2x\rangle$ is not of the form z_t for any $t < s$. *)
(i) $\text{content}(\alpha_s) \subseteq \text{content}(\tau_i) \cup W_{p_j}$ enumerated up to now;
(ii) $\langle j, 1, y_s\rangle \in W_{\mathbf{M}(\alpha_s, \tau_i' \#^{|\alpha_s| - |\tau_i'|})} - \text{content}(\alpha_s)$;
(iii) $\langle j, 1, y_s\rangle > \max(\text{content}(\alpha_s[k]))$, where $k = |\alpha_s| - 1$;
(iv) $\max(\text{content}(\alpha_s)) \geq \langle j, 1, y_s + 4\rangle$.
If and when such α_s, y_s are found, proceed to step 3.2.
3.2. Enumerate W_{p_j} enumerated until now except for $\langle j, 1, y_s\rangle$ into W_{q_j}.
Enumerate more and more of $\{\langle j, 1, 2x\rangle \mid 2x \neq y_s\}$ into W_{q_j}, until a $\gamma_s \supset \alpha_s$ and $z_s \in N$ are found such that:
(i) $\text{content}(\gamma_s) \subseteq \text{content}(\tau_i) \cup W_{q_j}$ (enumerated until then)
(ii) $z_s \in W_{\mathbf{M}(\gamma_s, \tau_i'\#^{|\alpha_s|-|\tau_i|}\text{‘no’}\#^{|\gamma_s|-|\alpha_s|-1})} \cap ((I_{\max(\text{content}(\gamma_s))} - (\text{content}(\gamma_s) \cup I_{\max(\text{content}(\alpha_s))} \cup W_{p_j}$ enumerated up to now$)) \cup \{\langle j, 1, y_s\rangle\})$.
(* Note that by considering z_s not to come from $I_{\max(\text{content}(\alpha_s))}$ (except for $\langle j, 1, y_s\rangle$), we have made sure that the answers to conjectures between τ_i (inclusive) and α_s (exclusive) are all #, as long as step 2 does not succeed. Furthermore we also ensured that W_{p_j} would not contain z_s, except for the case when $z_s = \langle j, 1, y_s\rangle$. *).
If and when such γ_s and z_s are found, proceed to step 3.3.
3.3 If $z_s = \langle j, 1, y_s\rangle$, then stop enumerating W_{q_j}, and wait until step 2 succeeds.
Else enumerate $\langle j, 1, r_s\rangle$ into W_{p_j}, where $r_s = y_s + 1$ or $r_s = y_s + 3$, and $\langle j, 1, r_s\rangle \neq z_s$, and proceed to step 3.4.
3.4 If there exists a $t \leq s$, $W_{\mathbf{M}(\alpha_t, \tau_i'\#^{|\alpha_t|-|\tau_i|})}$ enumerates $\langle j, 1, r_t\rangle$ within s steps, then wait until step 2 succeeds.
Otherwise proceed to stage $s + 1$.
End stage s

We now define τ_{i+1}, τ'_{i+1} based on a case analysis.
Case 1: Step 2 succeeds in exiting.
In this case let σ be as found in step 2 above.
Let σ' be an extension of τ_i' defined as follows. For $|\tau_i| \leq m \leq |\sigma|$, define

$$\sigma'(m) = \begin{cases} \#, & \text{if } W_{\mathbf{M}(\sigma[m], \sigma'[m])} \cap I_{\max(\text{content}(\sigma[m]))} \subseteq \text{content}(\tau_i) \cup W_{p_j}; \\ w, & \text{otherwise, where } w \text{ is the least element in} \\ & (W_{\mathbf{M}(\sigma[m], \sigma'[m])} \cap I_{\max(\text{content}(\sigma[m]))}) - (\text{content}(\tau_i) \cup W_{p_j}). \end{cases}$$

Note that the above answers/counterexamples (as given by σ' on input $\sigma\#$) are consistent with any language L such that $\text{content}(\tau_i) \cup (W_{p_j} \cap I_{\max(\text{content}(\sigma))}) \subseteq L \subseteq \text{content}(\tau_i) \cup W_{p_j}$. Furthermore, on some γ such that $\tau_i \subset \gamma \subseteq \sigma$, $\mathbf{M}$ does receive a ‘no’ answer (as it will do so on σ, if not before).

Now, let $\tau_{i+1} = \alpha \diamond \langle j, 1, 2y + 1\rangle$, where α is the smallest extension of $\sigma\#$ such that for some k, $|\sigma| + 1 \leq k \leq |\alpha|$,

(i) $\text{content}(\tau_i) \cup (W_{p_j} \cap I_{\max(\text{content}(\sigma))}) \cup (W_{p_j} - \{\langle j, 1, 2x + 1\rangle \mid x \in N\}) \subseteq \text{content}(\alpha) \subseteq \text{content}(\tau_i) \cup W_{p_j}$, and
(ii) $W_{\mathbf{M}(\alpha[k], \sigma'\#^{k-|\sigma'|})}$ contains an element in $I_{\max(\text{content}(\alpha[k]))} - \text{content}(\alpha)$, and
(iii) $\langle j, 1, 2y + 1\rangle \notin I_{\max(\text{content}(\alpha))}$ is a large number such that $W_{p_j} - I_{\max(\text{content}(\alpha))}$ contains an element smaller than $\langle j, 1, 2y + 1\rangle$.

(This is to ensure that W_{p_j} indeed satisfies condition (3B), and has an element of the form $\langle j, 1, 2x + 1\rangle$ which is smaller than maximum element in the diagonalizing language).

Note that there exists such a α (satisfying (i) and (ii)). To see this, suppose otherwise. Let $\gamma \supseteq \sigma\#$ be a $\mathbf{ResBNC}^{2n-2}\mathbf{Bc}$ ($\mathbf{ResBNC}^{2n-2}\mathbf{Ex}^*$)-locking sequence for $\mathbf{M}$ on $\text{content}(\tau_i) \cup W_{p_j}$, where the counterexample/answers beyond σ' are always # (note that if α, as claimed, does not exist, then there must exist such a locking sequence γ, as all the answers beyond $\sigma\#$ are always 'yes'). Without loss of generality assume that $\mathrm{content}(\gamma) \supseteq \mathrm{content}(\tau_i) \cup (W_{p_j} \cap I_{\max(\mathrm{content}(\sigma))}) \cup (W_{p_j} - \{\langle j, 1, 2x+1\rangle \mid x \in N\})$. Let $w > \max(\mathrm{content}(\gamma))$ be such that $\langle j, 1, 2w+3\rangle \in W_{p_j}$ and $W_{\mathbf{M}(\gamma, \sigma'\#^{|\gamma|-|\sigma'|})}$ contains $\langle j, 1, 2w+1\rangle$. Note that there exists such a w as γ is a locking sequence for $\mathbf{M}$ on $\mathrm{content}(\tau_i) \cup W_{p_j}$. Now taking $\alpha = \gamma\diamond\langle j, 1, 2w+3\rangle$ satisfies (i) and (ii) as $\langle j, 1, 2w+1\rangle \notin \mathrm{content}(\alpha)$, but $\langle j, 1, 2w+1\rangle \in W_{\mathbf{M}(\alpha, \sigma'\#^{|\alpha|-|\sigma'|})}$ (as γ was $\mathbf{ResBNC}^n\mathbf{Bc}$ ($\mathbf{ResBNC}^n\mathbf{Ex}^*$)-locking sequence for $\mathbf{M}$ on $\mathrm{content}(\tau_i) \cup W_{p_j}$).

So let α and τ_{i+1} be as claimed.

Define τ'_{i+1} as an extension of σ' such that for $|\sigma'| \le m < |\tau_{i+1}|$,

$$\tau'_{i+1}(m) = \begin{cases} \#, & \text{if } W_{\mathbf{M}(\tau_{i+1}[m], \tau'_{i+1}[m])} \cap I_{\max(\mathrm{content}(\tau_{i+1}[m]))} \subseteq \mathrm{content}(\tau_{i+1}); \\ w, & \text{otherwise, where } w \text{ is the least element in} \\ & W_{\mathbf{M}(\tau_{i+1}[m], \tau'_{i+1}[m])} \cap I_{\max(\mathrm{content}(\tau_{i+1}[m]))} - \mathrm{content}(\tau_{i+1}). \end{cases}$$

It is easy to verify that the invariants (A), (B) are maintained. Also invariant (C) is maintained as $\mathbf{M}$ would receive at least two counterexamples for conjectures between τ_i (inclusive) and τ_{i+1} (exclusive) for the language $\mathrm{content}(\tau_{i+1})$ (one at σ or before, and one between $\sigma\#$ and α, due to property (ii) in the definition of α). (D) follows easily from definition of τ'_{i+1}, and (E) holds as $(W_{p_j} - \{\langle j, 1, 2x+1\rangle \mid x \in N\}) \subseteq \mathrm{content}(\tau_{i+1})$, thus $W_{p_j} - \mathrm{content}(\tau_{i+1}) \subseteq \{\langle j, 1, 2x+1\rangle \mid x \in N\}$ and condition (3B) is satisfied (note that W_{p_j} is infinite).

Case 2: Not Case 1, and there exist infinitely many stages in step 3.

Let $L = \mathrm{content}(\tau_i) \cup W_{p_j}$. Now, W_{p_j} contains all elements of the form $\langle j, 1, r_s\rangle$. However, for all t, $W_{\mathbf{M}(\alpha_t, \tau'_i\#^{|\alpha_t|-|\tau_i|})}$ does not contain $\langle j, 1, r_t\rangle$ (otherwise, at some stage step 3.4 would have succeeded in finding such a t). Note here that $\bigcup_s \alpha_s = \bigcup_s \sigma_s$ is a text for L, and $\tau'_i\#^\infty$ is a valid sequence of answers/counterexamples to $\mathbf{M}$ on input $\bigcup_s \sigma_s$ as step 2 did not succeed. Thus, $\mathbf{M}$ does not $\mathbf{ResBNC}^{2n-2}\mathbf{Bc}$ ($\mathbf{ResBNC}^{2n-2}\mathbf{Ex}^*$)-identify $L \in \mathcal{L}$.

Case 3: Not Case 1, and Stage s starts but does not end.

Now consider the execution in stage s. We first claim that step 3.1 succeeds in finding α_s as required. To see this, suppose otherwise. Let $\gamma \supseteq \sigma_s$ be a $\mathbf{ResBNC}^{2n-2}\mathbf{Bc}$ ($\mathbf{ResBNC}^{2n-2}\mathbf{Ex}^*$)-locking sequence for $\mathbf{M}$ on $\mathrm{content}(\tau_i) \cup W_{p_j}$, where the counterexample/answers beyond τ'_i are always # (note that as step 2 did not succeed, there must exist such γ, as all the answers beyond τ_i are 'yes' whenever conjecture-subset questions are asked). Here without loss of generality we assume that $W_{\mathbf{M}(\gamma, \tau'_i\#^{|\gamma|-|\tau_i|})} \supseteq W_{p_j} - I_{\max(\mathrm{content}(\gamma))}$ (for $\mathbf{ResBNC}^{2n-2}\mathbf{Bc}$-learnability this clearly holds; for $\mathbf{ResBNC}^{2n-2}\mathbf{Ex}^*$-learnability we could just replace γ by some extension (contained in $W_{p_j} \cup \mathrm{content}(\tau_i)$) such that this property is satisfied). Let m be an even number which is bigger than $x_s + \max(\mathrm{content}(\gamma))$. Then $\gamma\diamond\langle j, 1, m+4\rangle$ would qualify for being α_s, as $\langle j, 1, m\rangle > \max(\mathrm{content}(\gamma))$ and γ had the locking sequence property as mentioned above (and thus, $W_{\mathbf{M}(\gamma\diamond\langle j,1,m+4\rangle, \tau'_i\#^{|\gamma|-|\tau_i|+1})}$ contained $\langle j, 1, m\rangle$) allowing one to take $y_s = m$ in step 3.1.

In a similar way one can argue that step 3.3 also is reached. (Here we will need to use W_{q_j} instead of W_{p_j} and use α_s instead of σ_s, and use $\tau'_i\#^{|\alpha_s|-|\tau_i|}$'no' instead of τ'_i in the previous argument about reaching step 3.2; rest of the argument is essentially the same).

So assume step 3.3 is reached and consider the following subcases.

SubCase 3.1: $z_s \neq \langle j, 1, y_s\rangle$.

Since stage s does not end, step 3.4 must have succeeded in finding a $t \le s$, such that $W_{\mathbf{M}(\alpha_t, \tau'_i\#^{|\alpha_t|-|\tau_i|})}$ enumerates $\langle j, 1, r_t\rangle$.

Fix such a t.

Let $X = (\mathrm{content}(\gamma_t) \cup W_{p_j}) - \{\langle j, 1, r_t\rangle\}$. Note that X does not contain z_t. Let $\tau_{i+1} = \alpha\#$, where α is an extension of γ_t such that $\mathrm{content}(\alpha) = X$.

Define τ'_{i+1} to be extension of $\tau'_i\#^{|\alpha_t|-|\tau_i|}\langle j, 1, r_t\rangle$ as follows. For $|\alpha_t| < m < |\tau_{i+1}|$

$$\tau'_{i+1}(m) = \begin{cases} \#, & \text{if } W_{\mathbf{M}(\tau_{i+1}[m], \tau'_{i+1}[m])} \cap I_{\max(\mathrm{content}(\tau_{i+1}[m]))} \subseteq \mathrm{content}(\tau_{i+1}); \\ w, & \text{otherwise, where } w \text{ is the least element in} \\ & W_{\mathbf{M}(\tau_{i+1}[m], \tau'_{i+1}[m])} \cap I_{\max(\mathrm{content}(\tau_{i+1}[m]))} - \mathrm{content}(\tau_{i+1}). \end{cases}$$

Note here that answers as given by τ'_{i+1} are correct on prefixes of α_t, as step 2 did not succeed and $\max(\text{content}(\alpha_t[|\alpha_t|-1])) < \langle j, 1, y_t\rangle < \langle j, 1, r_t\rangle$.

It is easy to verify that the invariants (A), (B) are maintained. Also invariant (C) is maintained as **M** would receive at least two counterexamples between τ_i (inclusive) and τ_{i+1} (exclusive) for the language content(τ_{i+1}) (one at α_t and one at γ_t or before). (D) follows easily from definition of τ'_{i+1}, and (E) holds as $W_{p_j} - \text{content}(\tau_{i+1})$ contains exactly $\langle j, 1, r_t\rangle$. Thus, condition (3B) in definition of $\mathcal{L}$ is satisfied.

SubCase 3.2: $z_s = \langle j, 1, y_s\rangle$.

In this case let $X = \text{content}(\tau_i) \cup W_{q_j}$. Let $\tau_{i+1} = \alpha\#$, where α is an extension of γ_s such that content$(\alpha) = X$.

Define τ'_{i+1} to be extension of $\tau'_i\#^{|\alpha_s|-|\tau_i|}\langle j, 1, y_s\rangle$ as follows. For $|\alpha_s| < m < |\tau_{i+1}|$

$$\tau'_{i+1}(m) = \begin{cases} \#, & \text{if } W_{\mathbf{M}(\tau_{i+1}[m], \tau'_{i+1}[m])} \cap I_{\max(\text{content}(\tau_{i+1}[m]))} \subseteq \text{content}(\tau_{i+1}); \\ w, & \text{otherwise, where } w \text{ is the least element in} \\ & W_{\mathbf{M}(\tau_{i+1}[m], \tau'_{i+1}[m])} \cap I_{\max(\text{content}(\tau_{i+1}[m]))} - \text{content}(\tau_{i+1}). \end{cases}$$

Note here that answers as given by τ'_{i+1} are correct on prefixes of α_s as step 2 did not succeed, and $\max(\text{content}(\alpha_s[|\alpha_s|-1])) < \langle j, 1, y_s\rangle$.

It is easy to verify that the invariants (A), (B) are maintained. Also invariant (C) is maintained as **M** would receive at least two counterexamples between τ_i (inclusive) and τ_{i+1} (exclusive) for the language content(τ_{i+1}) (one at α_s and one at γ_s or before). (D) follows easily from definition of τ'_{i+1}, and (E) holds as $W_{p_j} - \text{content}(\tau_{i+1})$ has exactly the element $\langle j, 1, y_s\rangle$, and condition (3B) is satisfied.

Above cases complete the construction of τ_{i+1}.

Now once τ_{n-1} has been defined, then we have that $2n-2$ counterexamples have already been provided to **M** based on τ'_{n-1}. Now, choose $j \notin S_{n-1}$. Let p_j, q_j be such that $W_{p_j} = \{\langle j, 0, \langle p_j, q_j\rangle\rangle\} \cup \{\langle j, 1, 2x+1\rangle \mid x \in N\}$.

Now **M** needs to **TxtBc***-identify content$(\tau_i) \cup W_{p_j}$ as well as content$(\tau_i) \cup \{\langle j, 1, 2x+1\rangle \mid x \leq w+1, x \neq w\}$, for all possible w, from any text extending τ_{n-1} without receiving any further counterexamples beyond τ_{n-1}, an impossible task by Proposition 11.

This proves the theorem. ∎

One can extend the above proof to show that $\mathbf{BNC}^n\mathbf{Ex} - \mathbf{ResBGNC}^{2n-2}\mathbf{Bc} \neq \emptyset$. The main problem to address is that in the search for α_s and γ_s, the learner may not be asking conjecture-subset questions, but still converge to a grammar for content$(\tau_i) \cup W_{p_j}$ and content$(\tau_i) \cup W_{q_j}$, in steps 3.1 and 3.2. To address this, do the last step (i.e., the set W_{p_j} used after τ_{n-1} is defined) first. That is, initially we (temporarily) assume that $W_{p_{sp}} = \{\langle 0, 0, \langle p_{sp}, q_{sp}\rangle\rangle\} \cup \{\langle 0, 1, 2x+1\rangle \mid x \in N\}$ is already a subset of the diagonalizing language. Correspondingly, we will look for counterexamples only outside $W_{p_{sp}}$. Furthermore, in step 3.1 we search for α_s such that (in addition to (i), (ii) and (iv) of step 3.1, where (i) now is updated to allow α_s to contain members of $W_{p_{sp}}$), there exists a k_s such that $\langle j, 1, y_s\rangle > \max(\text{content}(\alpha_s[k_s]))$, and $\mathbf{M}(\alpha_s[k], \tau'_i\#^{k-|\tau'_i|})$ asks a conjecture-subset question for $k = |\alpha_s|$, but does not ask a conjecture-subset question for $k_s < k < |\alpha_s|$. This is just to ensure similar properties as before when the first conjecture-subset question is asked by **M** beyond $\alpha_s[k_s]$. Update in step 3.2 is simpler as we just take care of $W_{p_{sp}}$ as mentioned above.

Analysis remains almost the same except that

- in the argument in case 1 for claiming that α exists, one now needs to consider the least extension of $\gamma\diamond\langle j, 1, 2w+3\rangle$ (containing elements only from content$(\gamma) \cup \{\langle j, 1, 2w+3\rangle\} \cup W_{p_{sp}}$) on which a question is asked.
- case 3, where we argue that step 3.3. is reached, needs to be modified. We again consider the locking sequence γ for content$(\tau_i) \cup W_{p_j} \cup W_{p_{sp}}$, and argue as follows. Let $X = W_{p_j} \cup \text{content}(\tau_i) \cup (W_{p_{sp}} \cap I_{\max(\text{content}(\gamma))}) \cup \{\langle 0, 1, 2*\max(\text{content}(\gamma))+3\rangle\}$, and H be an increasing text for X. Then, either no conjecture-subset question is asked by **M** beyond γ for the text $\gamma\diamond\langle j, 1, m+4\rangle\diamond H(0)\diamond\langle j, 1, m+4+6\rangle\diamond H(1)\diamond\langle j, 1, m+4+2*6\rangle H(2)\cdots$, (in which case the learner does not identify X which is in $\mathcal{L}$) or the first time beyond γ when a conjecture-subset question is asked, also gives us α_s fulfilling the requirements as in step 3.1. Similar (though simpler) argument works for the search of γ_s. We omit the details.

One can modify the above proof to show the following.

Theorem 26. *Suppose* $n, m \in N$. $\mathbf{BNC}^n\mathbf{Ex} - \mathbf{ResBNC}^{2n-2}\mathbf{Bc}^m \neq \emptyset$.

Above theorem can be proved by considering $m + 1$ elements $\langle j, 1, r_s^k\rangle$, $k \leq m$, instead of just $\langle j, 1, r_s\rangle$ as in the Proof of Theorem 25 (note that in step 3.1(iv), one would correspondingly need $\alpha_s(|\alpha_s|)$ to be larger than $\langle j, 1, y_s + 2m + 2\rangle$, so that we are able to use appropriate m values at step 3.3 Else clause.) We omit the details.

Interestingly, if we consider behaviorally correct learners that are allowed to make any finite number of errors in almost all correct conjectures, then n short (even least) counterexamples can be always substituted by just n 'no' answers. (For the model **NC**, the lower bound $2n - 1$ for the simulation by **Res**-type learners still holds even for $\mathbf{Bc}^*$-learnability, as shown in [10].)

Theorem 27. *For all* $n \in N$, $\mathbf{LBGNC}^n\mathbf{Bc}^* \subseteq \mathbf{ResBNC}^n\mathbf{Bc}^*$.

Proof. First note that one can simulate a $\mathbf{LBGNC}^n\mathbf{Bc}^*$-learner $\mathbf{M}$ by a $\mathbf{LBNC}^n\mathbf{Bc}^*$-learner $\mathbf{M}'$ as follows. If $\mathbf{M}(\sigma, \sigma')$ does not ask a conjecture-subset question, then $\mathbf{M}'(\sigma, \sigma')$ is a grammar for $W_{\mathbf{M}(\sigma,\sigma')} - \{x \mid x \leq \max(\text{content}(\sigma))\}$; otherwise $\mathbf{M}'(\sigma, \sigma') = \mathbf{M}(\sigma, \sigma')$. It is easy to verify that on any input text T, $\mathbf{M}'$ gets exactly the same counterexamples as $\mathbf{M}$ does, and all conjectures of $\mathbf{M}'$ are finite variants of corresponding conjectures of $\mathbf{M}$. Thus, any language $\mathbf{LBGNC}^n\mathbf{Bc}^*$-identified by $\mathbf{M}$ is $\mathbf{LBNC}^n\mathbf{Bc}^*$-identified by $\mathbf{M}'$.

Hence, it suffices to show $\mathbf{LBNC}^n\mathbf{Bc}^* \subseteq \mathbf{ResBNC}^n\mathbf{Bc}^*$.

Suppose $\mathbf{M}$ $\mathbf{LBNC}^n\mathbf{Bc}^*$-identifies $\mathcal{L}$. Define $\mathbf{M}'$ as follows. Suppose T is the input text.

The idea is for $\mathbf{M}'$ to output $\max(\text{content}(T[m]))+1$ variations of grammar output by $\mathbf{M}$ on $T[m]$. These grammars would be for the languages: $W_{\mathbf{M}(T[m])} - \{x \mid x \neq i \text{ and } x \leq \max(\text{content}(T[m']))\}$, where $T[m']$ is the input seen by $\mathbf{M}'$ when generating this ith variant (where $0 \leq i \leq \max(\text{content}(T[m]))$). These grammars would thus allow $\mathbf{M}'$ to determine the least counterexample, if any, that $\mathbf{M}$'s output on $T[m]$ would have generated.

Formally conjectures of $\mathbf{M}'$ will be of the form $P(j, m, i, s)$, where $W_{P(j,m,i,s)} = W_j - \{x \mid x \neq i \text{ and } x \leq s\}$.

We assume that $\mathbf{M}$ outputs grammar for $\emptyset$ until it sees at least one element in the input. This is to avoid having any counterexamples until input contains at least one element (which in turn makes the notation easier for the following proof).

On input $T[0]$, conjecture of $\mathbf{M}'$ is $P(\mathbf{M}(\Lambda, \Lambda), 0, 0, 0)$.

The invariant we will have is: If $\mathbf{M}'(T[m], T'[m]) = P(j, r, i, s)$, then, (i) $j = \mathbf{M}(T[r], T''[r])$, where $T''[r]$ is the sequence of least counterexamples for $\mathbf{M}$ on input $T[r]$ (for the language content(T)), (ii) $s = \max(\text{content}(T[m]))$, (iii) $r \leq m$, (iv) $i \leq \max(\text{content}(T[r]))$, and (v) $W_j - L$ does not contain any element $< i$. Invariant is clearly satisfied for $m = 0$.

Suppose $\mathbf{M}'(T[m], T'[m]) = P(\mathbf{M}(T[r], T''[r]), r, i, s)$. Then we define $\mathbf{M}'(T[m + 1], T'[m + 1])$ as follows.

If $T'(m)$ is 'no' answer, then let $T''(r) = i$, and let $\mathbf{M}'(T[m + 1], T'[m + 1]) = P(\mathbf{M}(T[r + 1], T''[r + 1]), r + 1, 0, \max(\text{content}(T[m + 1])))$.

Else if $i = \max(\text{content}(T[r]))$, then let $T''(r) = \#$, and let $\mathbf{M}'(T[m + 1], T'[m + 1]) = P(\mathbf{M}(T[r + 1], T''[r + 1]), r + 1, 0, \max(\text{content}(T[m + 1])))$.

Else, $\mathbf{M}'(T[m + 1], T'[m + 1]) = P(\mathbf{M}(T[r], T''[r]), r, i + 1, \max(\text{content}(T[m + 1])))$.

Now it is easy to verify that the invariant is maintained. It also follows that T'' constructed as above is correct sequence of least counterexamples for $\mathbf{M}$ on input T. Moreover, each restricted 'no' answer in T' corresponds to a least counterexample in T''. Thus, $\mathbf{M}'$ gets exactly as many counterexamples as $\mathbf{M}$ does, and $\mathbf{M}'$ conjectures are $*$-variants of the conjectures of $\mathbf{M}$ (except that each conjecture of $\mathbf{M}$ is repeated finitely many times by $\mathbf{M}'$, with finite variations). It follows that $\mathbf{M}'$ $\mathbf{ResBNC}^n\mathbf{Bc}^*$-identifies $\mathcal{L}$. ∎

Corollary 28. *For all* $n \in N$, $\mathbf{LBNC}^n\mathbf{Bc}^* = \mathbf{BNC}^n\mathbf{Bc}^* = \mathbf{ResBNC}^n\mathbf{Bc}^* = \mathbf{LBGNC}^n\mathbf{Bc}^* = \mathbf{BGNC}^n\mathbf{Bc}^* = \mathbf{ResBGNC}^n\mathbf{Bc}^*$.

Our next result in this section shows how **BNCBc**-learners using just answers 'yes' or 'no' can simulate $\mathbf{LBNCEx}^*$-learners getting unbounded number of negative answers/counterexamples.

Proposition 29. $\mathbf{LBNCEx}^* \subseteq \mathbf{ResBNCBc}$.

Proof. As $\mathbf{LBNCEx}^* = \mathbf{BNCEx}^*$ (see [11]) and $\mathbf{ResBNCBc} = \mathbf{BNCBc}$ (proof of $\mathbf{ResNCBc} = \mathbf{NCBc}$ in [11], shows this also) it suffices to show that $\mathbf{BNCEx}^* \subseteq \mathbf{BNCBc}$.

The idea is to patch the errors of omission by using the input text and to patch errors of commission by using the counterexamples (where we need to be somewhat careful for errors of commission which are larger than the largest element in the input). We now proceed formally.

Suppose **M** **BNCEx***-identifies $\mathcal{L}$. Define $\mathbf{M}'$ as follows. $\mathbf{M}'$ on input σ simulates **M**. (We will argue below that counterexamples for any conjectures of **M** are available to $\mathbf{M}'$ too, so the counterexample text for **M** can be created using the counterexample text for $\mathbf{M}'$.)

If **M** on input σ (with the appropriate counterexamples) outputs a grammar p, then $\mathbf{M}'$ outputs grammar $H(p, \sigma)$ defined as follows. Let S_p denote the set of counterexamples $\mathbf{M}'$ has received for the conjectures $H(p, \cdot)$ that $\mathbf{M}'$ has made up to now (note that p might have been output by **M** on some proper prefixes of σ too).

Let $I_m = \{x \mid x \le m\}$.

$$W_{H(p,\sigma)} = \text{content}(\sigma) \cup ((W_p \cap I_{\max(\text{content}(\sigma))}) - S_p) \cup (W_p - I_{\max(\text{content}(\sigma))}) \cap X_{p,|\sigma|},$$

where $X_{p,m}$ is N, if $\text{card}(W_p) \ge m$, and $\emptyset$ otherwise. Note that if **M** would have received a counterexample to its conjecture p, then either S_p is non-empty, or $\mathbf{M}'$ would also have received a counterexample to its conjecture $H(p, \sigma)$. Thus counterexample text for **M** can be constructed by $\mathbf{M}'$.

We now argue that $\mathbf{M}'$ would **BNCBc**-identify $\mathcal{L}$. Let T be the input text for $L \in \mathcal{L}$. Suppose T' is the counterexample text prepared for **M** by $\mathbf{M}'$ in the above simulation. Then, clearly $\mathbf{M}(T, T')$ would converge to some grammar p which is a finite variant of L. Now if L is finite, then W_p is also finite. Thus, for all but finitely many initial segments of T, $\mathbf{M}'$ would output a grammar for $W_{H(p,\sigma)} = \text{content}(\sigma) \cup ((W_p \cap I_{\max(\text{content}(\sigma))}) - S_p)$ (as $X_{p,m}$ is empty for all but finitely many m). Thus, all the errors of omission of W_p as well as any errors of commission are patched (errors of commission which are bigger than max(content(σ)) are clearly not output; errors of commission which are smaller than max(content(σ)) eventually go into S_p and are thus patched too).

If L is infinite, then all the errors of omission of W_p as well as any errors of commission are patched (all errors of commission in this case eventually go into S_p).

It follows that $\mathbf{M}'$ **BNCBc**-identifies $\mathcal{L}$. ∎

Proposition 30 (*Based on [5]*). *Suppose X is an infinite language, and S is a finite subset of X. Suppose $n \in N$. Then $\mathcal{L} = \{S \subseteq L \subseteq X \mid \text{card}(X - L) \le 2n + 1\} \notin$* $\mathbf{TxtBc}^n$.

Theorem 31. *For all $m, n \in N$,*

(a) $\mathbf{TxtEx}^{2n+1} - \mathbf{LBGNC}^m\mathbf{Bc}^n \neq \emptyset$.
(b) $\mathbf{TxtEx}^{n+1} - \mathbf{LBGNC}^m\mathbf{Ex}^n \neq \emptyset$.
(c) $\mathbf{ResBNC}^m\mathbf{Ex}^{2n} \subseteq \mathbf{ResBNC}^m\mathbf{Bc}^n$.
(d) $\mathbf{BNC}^m\mathbf{Ex}^{2n} \subseteq \mathbf{BNC}^m\mathbf{Bc}^n$.
(e) $\mathbf{LBNC}^m\mathbf{Ex}^{2n} \subseteq \mathbf{LBNC}^m\mathbf{Bc}^n$.
(f) $\mathbf{ResBGNC}^m\mathbf{Ex}^{2n} \subseteq \mathbf{ResBGNCBc}^n$.
(g) $\mathbf{BGNC}^m\mathbf{Ex}^{2n} \subseteq \mathbf{BGNC}^m\mathbf{Bc}^n$.
(h) $\mathbf{LBGNC}^m\mathbf{Ex}^{2n} \subseteq \mathbf{LBGNC}^m\mathbf{Bc}^n$.

Proof. (a) Let $\mathcal{L} = \{L \mid m \le \text{card}(N - L) \le m + 2n + 1\}$. It is easy to verify that $\mathcal{L} \in \mathbf{TxtEx}^{2n+1}$ (one eventually outputs a grammar for $N - S$, where S is the set of least m elements missing from the input). Suppose by way of contradiction that **M** $\mathbf{LBGNC}^m\mathbf{Bc}^n$-identifies $\mathcal{L}$. Define σ_i, σ_i', $i \le m$, by induction on i, as follows.

$$\sigma_0 = \sigma_0' = \Lambda.$$

By induction we will have the invariants that answers given by σ_i' on σ_i are consistent with any L such that $\text{content}(\sigma) \subseteq L \subseteq N - \text{content}(\sigma_i')$. Furthermore, card(content(σ_i')) is at least i.

Now let $\sigma_{i+1} = \sigma\#$, where σ is the smallest extension, if any, of σ_i such that $\mathbf{M}(\sigma, \sigma_i'\#^{|\sigma|-|\sigma_i'|})$ asks a conjecture-subset question and $W_{\mathbf{M}(\sigma,\sigma_i'\#^{|\sigma|-|\sigma_i'|})} - \text{content}(\sigma)$ contains an element in $I_{\max(\text{content}(\sigma))}$. If σ_{i+1} gets defined, then $\sigma_{i+1}' = \sigma_i'\#^{|\sigma|-|\sigma_i|}z$, where $z = \min(W_{\mathbf{M}(\sigma,\sigma_i'\#^{|\sigma|-|\sigma_i'|})} - \text{content}(\sigma))$. It is easy to verify that the invariants are satisfied. Now, let $r \le m$ be maximum such that σ_r is defined. Then for any extension σ of σ_r, such that $\text{content}(\sigma) \subseteq N - \text{content}(\sigma_r')$, **M** gets '#' answers (as either it does not ask conjecture-subset question or

$W_{\mathbf{M}(\sigma,\sigma'_r\#^{|\sigma|-|\sigma'_r|})} - \text{content}(\sigma)$ does not contain an element in $I_{\max(\text{content}(\sigma))}$). Thus, now $\mathbf{M}$ needs to $\mathbf{TxtBc}^n$-identify all languages in $\mathcal{L}$ which contain content(σ_r) but do not contain content(σ'_r), an impossible task by Proposition 30.

(b) can be proved similar to part (a).

(c–h) This proof is based on [5] proof of $\mathbf{TxtEx}^{2n} \subseteq \mathbf{TxtBc}^n$ (see [12] for a proof). We give the details for completeness. Suppose $\mathbf{M}$ $\mathbf{ResBNCEx}^{2n}$-identifies $\mathcal{L}$. Define $\mathbf{M}'$ as follows.

Let $P(e, A, B)$ be such that $W_{P(e,A,B)} = A \cup (W_e - S)$, where S is the set of least n elements in $W_e - B$ (if $W_e - B$ does not contain at least n elements, then we just take S to be $W_e - B$). By induction on length of input, it will be easy to verify that $\mathbf{M}'$ receives exactly the same counterexamples at exactly the same inputs as $\mathbf{M}$ does (for $\mathbf{GNC}$-models, $\mathbf{M}'$ asks questions on the same inputs as $\mathbf{M}$ does). Now on input (σ, σ'), if $\mathbf{M}'$ has already received m counterexamples/'no' answers, then $\mathbf{M}'$ outputs $P(\mathbf{M}(\sigma, \sigma'), \text{content}(\sigma), \text{content}(\sigma))$. Otherwise, $\mathbf{M}'$ outputs $P(\mathbf{M}(\sigma, \sigma'), \text{content}(\sigma), I_{\max(\text{content}(\sigma))})$.

It is easy to verify that $\mathbf{M}'$ receives exactly the same counterexample sequence as $\mathbf{M}$ (as before getting m counterexamples, the grammar output by $\mathbf{M}'$ enumerates the same elements in $I_{\max(\text{content}(\sigma))} - \text{content}(\sigma)$, as enumerated by the grammar output by $\mathbf{M}$). Now consider any text T for a language $L \in \mathcal{L}$, with T' being corresponding sequence of counterexamples. Suppose $\mathbf{M}(T, T')$ converges to e. Let $S' = W_e - L$. Suppose t is such that

(i) $\mathbf{M}(T[t], T'[t]) = e$, for all $t' \geq t$,
(ii) $L - W_e \subseteq \text{content}(T[t])$,
(iii) $T'(x) = \#$, for all $x \geq t$, and
(iv) for all $x \leq \max(S')$, if $x \in L$, then $x \in \text{content}(T[t])$.

Now, consider the following cases.

Case 1: $W_e - L$ contains at least n elements.

In this case, for all $t' \geq t$, S as computed by $P(\mathbf{M}(T[t'], T'[t']), \text{content}(T[t']), B)$, (where $B = \text{content}(\sigma)$ or $I_{\max(\text{content}(\sigma))}$, based on whether $\mathbf{M}'$ gets m or smaller number of counterexamples), consists of least n elements in S'. Furthermore, $L - W_e \subseteq \text{content}(T[t'])$. Thus, $\text{card}(W_{\mathbf{M}(T[t'],T'[t']}\mathbf{\Delta} L) = \text{card}(S') - n \leq n$.

Case 2: $W_e - L$ contains $<n$ elements.

In this case, for all $t' \geq t$, S as computed by $P(\mathbf{M}(T[t'], T'[t']), \text{content}(T[t']), B)$, (where $B = \text{content}(\sigma)$ or $I_{\max(\text{content}(\sigma))}$, based on whether $\mathbf{M}'$ gets m or smaller number of counterexamples), is a superset of S'. Furthermore, $L - W_e \subseteq \text{content}(T[t'])$. Thus, $\text{card}(W_{\mathbf{M}(T[t'],T'[t']}\mathbf{\Delta} L) \leq n - \text{card}(S') \leq n$.

In either case, $\mathbf{M}'$ would $\mathbf{Bc}^n$-identify the input (in appropriate counterexample model). ∎

6. Effects of counterexamples being constrained/not-constrained to be short

In this section we explore how, within the framework of our models, short counterexamples fair against arbitrary or least counterexamples (this includes also the cases when just answers 'no' are returned instead of counterexamples).

First, we use a result from [10] to establish that one answer 'no' used by an $\mathbf{NCEx}$-learner can sometimes do more than unbounded number of least (short) counterexamples used by $\mathbf{Bc}^*$-learners.

Theorem 32 (*[10]*). $\mathbf{ResNC^1Ex} - \mathbf{LBGNCBc}^* \neq \emptyset$.

(Jain and Kinber [11] actually showed $\mathbf{ResNC^1Ex} - \mathbf{LBNCBc}^* \neq \emptyset$, however the above result follows as for unbounded number of counterexamples, $\mathbf{GNC}$-model does not give any advantage over $\mathbf{NC}$-model).

Note that the advantages of least examples/counterexamples in speeding up learning has been studied in other situations also, such as learning of non-erasing pattern languages [19]. However, in our model of $\mathbf{BNC}$-learning versus $\mathbf{LNC}$-learning, the $\mathbf{LNC}$-learner does get least counterexamples, and $\mathbf{BNC}$-learner gets just a counterexample, if there exists one below the maximal positive data seen so far. This seems on the surface to hurt, as $\mathbf{BNC}$-learner is likely to get less (negative) data. In fact, that is the case as Jain and Kinber [11] showed that, for $a \in N \cup \{*\}$, for $\mathbf{I} \in \{\mathbf{Ex}^a, \mathbf{Bc}^a\}$, $\mathbf{LBNCI} \subset \mathbf{ResNCI}$. However, when we consider counting/bounding, there is a *charge* for every counterexample. Consequently, a $\mathbf{BNC}$-learner is not being charged for (unnecessary) negative data, if it does not receive it. As a result, the possibility of getting negative data which are $\leq$ maximal positive data seen in the input so far can be turned to an advantage – in terms of cost of learning. This is what is exploited in getting the following result. It shows that one short counterexample can sometimes give a learner more than any bounded number of least counterexamples (perhaps, it would be interesting to explore if there exist practically interesting classes of concepts – say, patterns of finite automata/regular expressions – where a similar effect of saving a number of *short* counterexamples for overinclusive conjectures in a query learning model over a number of *least* counterexamples would take place). The proof features an **Ex**-learner using just one bounded negative answer that cannot be simulated by an $\mathbf{LNC}^n\mathbf{Bc}^*$-learner for any n.

Theorem 33. *For all* $n \in N$, $\mathbf{ResBNC}^1\mathbf{Ex} - \mathbf{LGNC}^n\mathbf{Bc}^* \neq \emptyset$.

Proof. Assume without loss of generality that $\langle \cdot, \cdot \rangle$ is monotonically increasing in both its arguments. Note that this implies $\langle i, 0\rangle \geq i$.

Let $A_k^j = \{\langle k, x\rangle \mid x \leq j\}$.

Let
$\mathcal{L} = \{L \mid (\exists S \mid \mathrm{card}(S) < \infty)(\exists f : S \to N)[$
1. $[k, k' \in S \wedge k < k'] \Rightarrow [\langle k, f(k)\rangle < \langle k', 0\rangle] \wedge$
2. $[L = \mathrm{CYL}_{\max(S)} \cup \bigcup_{k \in S - \max(S)} A_k^{f(k)}$ or
$L = \{\langle \max(S), f(\max(S)+2)\rangle\} \cup \bigcup_{k \in S} A_k^{f(k)}]$
$]\}$.

Intuitively, $\mathcal{L}$ above consists of some initial portions of cylinders, plus maybe a full cylinder. The cylinders are placed far apart from each other to avoid interference: maximal element in a cylinder with smaller index is smaller than the minimal element of a cylinder with larger index. This allows for learning with at most one bounded negative counterexample. However, using technique similar to that used in Proposition 11, for unconstrained counterexamples, one can show that a learner needs arbitrarily large number of counterexamples to learn the above class. We now proceed formally.

To see that $\mathcal{L} \in \mathbf{ResBNC}^1\mathbf{Ex}$ consider the following learner. On input σ, if no 'no' answers are yet received, then the learner first computes $k = \max(\{j \mid \langle j, x\rangle \in \mathrm{content}(\sigma)\})$. Then it outputs a grammar for $L = \mathrm{CYL}_k \cup (\mathrm{content}(\sigma) - \mathrm{CYL}_k)$. If there is a 'no' answer which has been received, then the learner outputs a grammar for $\mathrm{content}(\sigma)$. It is easy to verify that the above learner $\mathbf{ResBNC}^1\mathbf{Ex}$-identifies $\mathcal{L}$.

Now suppose by way of contradiction that some $\mathbf{M}$ $\mathbf{LGNC}^n\mathbf{Bc}^*$-identifies $\mathcal{L}$. Let $\sigma_0 = \sigma_0' = \Lambda$, $k_0 = 0$. Inductively define $\sigma_{i+1}, \sigma_{i+1}', f(k_i), k_{i+1}$ (for $i < n$) as follows.

Let σ be smallest extension of σ_i, if any, such that $\mathrm{content}(\sigma) \subseteq \mathrm{CYL}_{k_i} \cup \bigcup_{i'<i} A_{k_{i'}}^{f(k_{i'})}$ and $\mathbf{M}$ asks a conjecture-subset question on $(\sigma, \sigma_i'\#^{|\sigma|-|\sigma_i|})$ and $W_{\mathbf{M}(\sigma, \sigma_i'\#^{|\sigma|-|\sigma_i|})}$ contains an element which is not in $\mathrm{CYL}_{k_i} \cup \bigcup_{i'<i} A_{k_{i'}}^{f(k_{i'})}$ or is larger than $\max(\mathrm{content}(\sigma))$.

If there is such a σ, then let $\sigma_{i+1} = \sigma\#$, and $\sigma_{i+1}' = \sigma_i'\#^{|\sigma|-|\sigma_i|}w$ (where w is the least element in $W_{\mathbf{M}(\sigma, \sigma_i'\#^{|\sigma|-|\sigma_i|})}$ which is not in $\mathrm{CYL}_{k_i} \cup \bigcup_{i'<i} A_{k_{i'}}^{f(k_{i'})}$ or is larger than $\max(\mathrm{content}(\sigma))$). Let $f(k_i) = \max(\{y \mid \langle k_i, y\rangle \in \mathrm{content}(\sigma)\})$. Let k_{i+1} be such that $k_{i+1} > \langle k_i, f(k_i)\rangle$ and no element from $\mathrm{CYL}_{k_{i+1}}$ is present in $\mathrm{content}(\sigma_{i+1}')$.

Let m be the largest value such that σ_m, σ_m' are defined above. Now, $\mathbf{M}$ has to $\mathbf{TxtBc}^*$-identify both $\mathrm{CYL}_{k_m} \cup \bigcup_{i<m} A_{k_m}^{f(k_m)}$ and $A_{k_m}^r \cup \{\langle k_m, r+2\rangle\} \cup \bigcup_{i<m} A_{k_i}^{f(k_i)}$, for all possible r, without any further counterexamples, an impossible task by Proposition 11. ∎

The above is the strongest possible result, as **ResNCI** contains **LBNCI** (as shown in [11]).

We now consider the complexity (mind change [6]) advantages of having only short counterexamples. For this purpose, we need to modify the definition of learner slightly, to avoid biasing the number of mind changes. (This modification is used only for the rest of the current section).

Definition 34. A learner is a mapping from SEQ to $N \cup \{?\}$.

A learner $\mathbf{M}$ $\mathbf{TxtEx}_n$-identifies $\mathcal{L}$, iff it $\mathbf{TxtEx}$-identifies $\mathcal{L}$, and for all texts T for $L \in \mathcal{L}$, $\mathrm{card}(\{m \mid ? \neq \mathbf{M}(T[m]) \neq \mathbf{M}(T[m+1])\})$ is bounded by n.

One can similarly define the criteria with mind change bounds for learners receiving counterexamples.

Our next result demonstrates that there exists a **TxtEx**-learnable class (that is, learnable just from positive data – without any subset queries) that can be learned by a $\mathbf{BNC}^1\mathbf{Ex}$-learner using just *one negative answer* and at most one mind change and cannot be learned by **Ex**-learners using any number of arbitrary counterexamples and any bounded number of mind changes. It underscores the great power of even very limited negative data in learning processes in the limit

Theorem 35. *There exists a $\mathcal{L}$ such that*

(a) $\mathcal{L} \in \mathbf{ResBNC}^1\mathbf{Ex}_1$.
(b) $\mathcal{L} \in$ **TxtEx**, *and thus in* **NCEx** *and* **GNCEx**.
(c) *For all* $m \in N$, $\mathcal{L} \notin \mathbf{NCEx}_m$.
(d) *For all* $m \in N$, $\mathcal{L} \notin \mathbf{GNCEx}_m$.

Proof. Let $L_n = \{x \mid x < n \text{ or } x = n + 1\}$.

Let $\mathcal{L} = \{L_n \mid n \in N\}$.

Consider the following learner. Initially output a grammar for N. If and when a 'no' answer is received, output a grammar for L_n, where $n + 1$ is the maximal element in the input received. It is easy to verify that the above learner $\mathbf{ResBNC}^1\mathbf{Ex}_1$-identifies $\mathcal{L}$.

It is also easy to verify that $\mathcal{L} \in$ **TxtEx** as one could output, in the limit on text T, a grammar for L_n, for the least n such that $n \notin \text{content}(T)$.

We now show that $\mathcal{L} \notin \mathbf{NCEx}_m$. As the number of counterexamples are not bounded, it follows that $\mathcal{L} \notin \mathbf{GNCEx}_m$.

Suppose that by way of contradiction, $\mathbf{M}$ $\mathbf{NCEx}_m$-identifies $\mathcal{L}$. Then consider the following strategy to construct a diagonalizing language.

We will construct the diagonalizing language in stages. Construction is non-effective. We will try to define l_s and u_s, and segments σ_s, σ'_s (σ'_s is the sequence of counterexamples), for $s \le m + 1$.

The following invariants will be satisfied.

(A) $u_s - l_s = 4^{m+3-s}$.
(B) $\mathbf{M}$ on proper prefixes of σ_s has made s different conjectures.
(C) $\text{content}(\sigma_s) \subseteq \{x \mid x < l_s\}$.
(D) None of the conjectures made by $\mathbf{M}$ on proper prefixes of σ_s are for the language L_r, for $l_s \le r \le u_s$.
(E) $|\sigma'_s| = |\sigma_s|$.
(F) For $r < |\sigma_s|$, $\sigma'_s(r) = \#$, implies $W_{\mathbf{M}(\sigma_s[r],\sigma'_s[r])} \subseteq \{x \mid x < l_s\}$.
(G) For $r < |\sigma_s|$, $\sigma'_s(r) \ne \#$, implies $\sigma'_s(r) \in W_{\mathbf{M}(\sigma_s[r],\sigma'_s[r])}$, and $\sigma'_s(r) > u_s + 1$.

Initially, we let $l_0 = 0$ and $u_0 = l_0 + 4^{m+3}$, and $\sigma_0 = \sigma'_0 = \Lambda$. Note that the invariants are satisfied.

Stage s (for $s = 0$ to $s = m$)

1. Let T be a text for L_{l_s} which extends σ_s.
2. Let $t \ge |\sigma_s|$, be the least value, if any, such that $\mathbf{M}(T[t], T'[t])$ is a conjecture different from any conjecture $\mathbf{M}(T[w], T'[w])$, for $w < |\sigma_s|$, where

$$T'(w) = \begin{cases} \sigma'_s(w), & \text{if } w < |\sigma_s|; \\ \#, & \text{if } w \ge |\sigma_s| \text{ and } \mathbf{M}(T[w], T'[w]) = ?; \\ T'(r), & \text{if } w \ge |\sigma_s| \text{ and } \mathbf{M}(T[w], T'[w]) = \mathbf{M}(T[r], T'[r]), \\ & \text{for some } r < |\sigma_s|. \end{cases}$$

(* Note that, in this step, we do not need definition of $T'(w)$ when $\mathbf{M}(T[w], T'[w])$ makes a new conjecture at or beyond σ_s. For first such w (which is t found above) $T'(w)$ will be defined below). *)

If and when such a t is found, proceed to step 3.

3. Suppose $j = \mathbf{M}(T[t], T'[t])$.

If W_j contains an element $z \ge l_s + \frac{3(u_s - l_s)}{4}$, then

Let $l_{s+1} = l_s + \frac{u_s - l_s}{4}$.

Let $u_{s+1} = l_s + \frac{2(u_s - l_s)}{4}$.

Let $\sigma_{s+1} = T[t]\#$.
Let $\sigma'_{s+1} = T'[t]z$.
(* Note thus that $\mathbf{M}(T[t], T'[t])$ is not a correct grammar for L_r, where $l_{s+1} \leq r \leq u_{s+1}$. *)
Else,
Let $l_{s+1} = l_s + \frac{3(u_s - l_s)}{4}$.
Let $u_{s+1} = u_s$.
Let $\sigma_{s+1} = T[t]\#$.
Let $\sigma'_{s+1} = T'[t]\#$.
(* Note thus that $\mathbf{M}(T[t], T'[t])$ is not a correct grammar for L_r, where $l_{s+1} \leq r \leq u_{s+1}$. *)
End stage s

It is easy to verify that the invariants are satisfied. (A) clearly holds by definition of l_{s+1} and u_{s+1} in step 3. (B) holds as one extra new conjecture is found at stage s, before proceeding to stage $s+1$. (C) holds, as $l_{s+1} \geq l_s + \frac{u_s - l_s}{4} > l_s + 2$, and content($T$) as defined in step 1 is a subset of L_{l_s}. (D) holds by induction, and noting that the conjecture at $T[t]$ as found in step 2 of stage s, is made explicitly wrong by appropriate choices of l_{s+1} and u_{s+1} in step 4. (E) easily holds by construction. (F) and (G) hold by the definition of σ'_{s+1} at step 3.

Now, if step 2 does not succeed at a stage $s \leq m$, then clearly **M** does not **NCEx**-identify L_{l_s}. On the other hand if stage m does complete then **M** has already made $m + 1$ different conjectures (and thus at least m mind changes) on prefixes of σ_{m+1}, which are not grammars for $L_{l_{m+1}}$. Thus, **M** cannot $\mathbf{NCEx}_m$-identify $L_{l_{m+1}}$. ∎

Let $X = \{x \mid x > 0\}$. If we consider the class $\mathcal{L} = \{L_n \mid n > 0\} \cup \{X\}$, then we can get the above result using *class preserving* learnability (that is, the learner always uses grammars from the numbering defining the target class of languages for its conjectures, see [20] for formal definition) for $\mathbf{ResBNC^1Ex}$.

Theorem 36. *For all $m \in N$,*

(a) $\mathbf{LBNCEx}_m \subseteq \mathbf{LNC}^m\mathbf{Ex}_m$.
(b) $\mathbf{LBGNCEx}_m \subseteq \mathbf{LGNC}^m\mathbf{Ex}_m$.

Proof. We only show part (a). Part (b) can be done similarly.

Suppose **M** $\mathbf{LBNCEx}_m$-identifies $\mathcal{L}$. On input σ, $\mathbf{M}'$ simulates **M**, providing it with counterexample z for a grammar p iff $z \leq \max(\text{content}(\sigma))$ and $\mathbf{M}'$ itself had earlier received such a counterexample z for grammar p. Then, $\mathbf{M}'$ outputs the latest conjecture of **M**, if $\mathbf{M}'$ has not as yet received any counterexample for this conjecture (otherwise $\mathbf{M}'$ just outputs ?).

It is easy to verify that $\mathbf{M}'$ $\mathbf{LNC}^m\mathbf{Ex}_m$-identifies $\mathcal{L}$ – the number of mind changes is bounded by the number of mind changes of **M**, and the number of counterexample received is at most one per conjecture (with none for the final conjectures). Theorem follows. ∎

7. Comparison of learning via limited number of short counterexamples and finite number of queries

In this section we compare capabilities of **BGNC**- and **BNC**-learners with the learners using a finite number of subset, equivalence and superset queries returning counterexamples of arbitrary or least size or just answers 'yes' or 'no' (as it was established in [10], bounded number of negative answers to such queries returning *short* counterexamples does not add any advantages to **TxtEx**- or **TxtBc**-learners, even if a finite number of errors are allowed in the final correct conjectures).

7.1. Query models versus short negative counterexamples

First, we refer to some facts established in [10].

Theorem 37 *([10]). For* $\mathbf{I} \in \{\mathbf{ResSubQ^1Ex}, \mathbf{ResNC^1Ex}, \mathbf{ResEquQ^1Ex}\}$, $\mathbf{I} - \mathbf{LBNCBc}^* \neq \emptyset$.

As $\mathbf{LBNCBc}^* = \mathbf{LGNCBc}^*$, we immediately have $\mathbf{I} - \mathbf{LGNCBc}^* \neq \emptyset$, for $\mathbf{I} \in \{\mathbf{ResSubQ^1Ex}, \mathbf{ResNC^1Ex}, \mathbf{ResEquQ^1Ex}\}$.

For the superset queries, one can only get a slightly weaker result: learners using just one query of this type and getting answer ‘yes’ or ‘no’ can sometimes do better than **GNCBc**-learners making just bounded number of errors in almost all correct conjectures.

Theorem 38. *Suppose* $n, m \in N$. $\mathbf{ResSupQ^1Ex} - \mathbf{LBGNC^nBc^m} \neq \emptyset$.

Proof. Proof of $\mathbf{ResSupQ^1Ex} - \mathbf{LNC^nBc^m} \neq \emptyset$ (based on cylinderification of class in $\mathbf{ResSupQ^1Ex} - \mathbf{LNC^nBc} \neq \emptyset$) in [10] can easily be modified to give this result. ∎

Also, one superset query can sometimes do better than **Bc** or $\mathbf{Ex}^*$-learners using unbounded number of short least counterexamples.

Theorem 39. $\mathbf{ResSupQ^1Ex} - \mathbf{LBNCBc} \neq \emptyset$.

Proof. Jain and Kinber [10] showed $\mathbf{ResSupQ^1Ex} - \mathbf{LNCBc} \neq \emptyset$. As $\mathbf{LBNCBc} \subseteq \mathbf{LNCBc}$, theorem follows. ∎

As $\mathbf{LBNCBc} = \mathbf{LBGNCBc}$, we also have $\mathbf{ResSupQ^1Ex} - \mathbf{LBGNCBc} \neq \emptyset$.

Corollary 40. $\mathbf{ResSupQ^1Ex} - \mathbf{LBNCEx^*} \neq \emptyset$.

Note that Theorem 39 cannot be strengthened as Theorem 42 below shows. (Here also note that $\mathbf{LSupQ^*Bc^*} \subseteq \mathbf{TxtBc^*}$ [10].)

To prove our next result, Theorem 42, we need the following technical lemma.

Lemma 41. *Suppose* **M** $\mathbf{SupQ^*Bc^*}$*-identifies* $\mathcal{L}$*, and* $N \in \mathcal{L}$*. Then, there exists a finite set* S_N *such that for all* $L \in \mathcal{L}$*,* $S_N \subseteq L \Rightarrow L =^* N$ *– in particular,* L *is infinite.*

Proof. Let **M** be as in the hypothesis of the lemma, and σ be a $\mathbf{SupQ^*Bc^*}$-locking sequence for **M** on N (i.e., for any τ such that $\sigma \subseteq \tau$, (i) **M** does not ask any questions beyond σ on τ, and (ii) **M** on τ outputs a grammar for finite variant of N.)

Let $S_N = \text{content}(\sigma) \cup \{x \mid x$ is the counterexample provided to a question of **M** on a prefix of σ, when learning the language $N\}$. Now let $L \supseteq S_N$ be a member of $\mathcal{L}$. Then, for any text T for L, which extends σ, by hypothesis about σ, we have that **M** does not ask any questions beyond σ, and only outputs grammars for a finite variant of N. As **M** $\mathbf{SupQ^*Bc^*}$-identifies L, lemma follows. ∎

The next theorem shows that $\mathbf{ResBNCBc^1}$-learners, making just one error in almost all correct conjectures and using a finite number of negative short counterexamples, can simulate any $\mathbf{Bc^m}$-learner using a finite number of superset queries. Intuitively, the technique used is similar to that of showing that the class of infinite r.e. sets can be $\mathbf{BNCBc^1}$-learnt [11]. Additionally, the conjectures of the **SupQ**-learner are used to handle finite sets, by outputting the input data (plus one negative element), when the **SupQ**-learner conjectures finite sets – here one needs to carefully search for and verify the answers to the **SupQ**-learner.

Theorem 42. *Suppose* $m \in N$. $\mathbf{SupQ^*Bc^m} \subseteq \mathbf{ResBNCBc^1}$.

Proof. Suppose **M** $\mathbf{SupQ^*Bc^m}$-identifies $\mathcal{L}$. If $N \in \mathcal{L}$, then let S_N be as given by Lemma 41. Otherwise let $S_N = N$.

Define $\mathbf{M}'$ as follows (note that definition of $\mathbf{M}'$ depends on S_N, and thus is not effective in **M**). We will define $\mathbf{M}'$ as just outputting a sequence of conjectures on input T and receiving answers of yes/no for each of its conjectures being subset/not subset of input (restricted to maximum element of the input data). Let (Q_1^q, Q_2^q), $q \in N$, be an ordering of all pairs of finite sets such that each pair of finite sets appears infinitely often in the ordering. Intuitively, each pair is a guess at the set of questions asked by **M** on input T which are to be answered as yes/no for the input language.

$\mathbf{M}'(T)$
Let $p = 0, q = 0$.
Stage s
1. If no ‘no’ answer has yet been received, then
    If $S_N \not\subseteq \text{content}(T[s])$ and $\text{content}(T[s])$ is an initial segment of N, then output a grammar for $\text{content}(T[s])$ and go to stage $s + 1$.
    Else output a grammar for N and go to stage $s + 1$.

Else let z be the least element not present in content($T[s]$), and go to step 2.

2. (* Here we know that the input language is not N, and it seems that the least missing datum is z. *)
For each $j \in Q_2^q$, let x_j be least element such that $x_j \in \text{content}(T[s]) - W_{j,s}$ (if there is no such x_j for some $j \in Q_2^q$, then go to stage $s+1$, with value of $q = q+1$, and p unchanged).
Dovetail steps 3 and 4.
3. If it is ever found that $z \in \text{content}(T)$ or $x_j \in W_j$ for some $j \in Q_2^q$ or $\mathbf{M}(T)$ (in the simulation at step 4) asks a question beyond $T[s]$ or asks a question of the form 'is $W_j \supseteq L$', for $j \notin Q_1^q \cup Q_2^q$, then stop step 4 and go to stage $s+1$, with $q = q+1$ and p unchanged.
4. Below let g_t denote the conjecture output by $\mathbf{M}(T[t])$, where questions of the form 'is $W_j \supseteq L$' for $j \in Q_1^q$ are given 'yes' answers, and questions of the form 'is $W_j \supseteq L$' for $j \in Q_2^q$ are given 'no' answers with counterexample x_j. (If $\mathbf{M}$ asks a question outside $Q_1^q \cup Q_2^q$, then step 3 would eventually force the construction to go to stage $s+1$.)
Go to substage s.
Substage t
4.1 if $content(T[t+1]) \neq content(T[t])$, then
Output p. If W_p as a conjecture of $\mathbf{M}'$ generates an answer 'no', then go to stage $s+1$, with $p = p+1$ and q unchanged.
4.2 Output a grammar for the language A, where:

$$A = \begin{cases} \text{content}(T[t]), & \text{if content}(T[t]) \not\subseteq \bigcap_{j \in Q_1^q} W_j; \\ \text{content}(T[t]) \cup \{z\}, & \text{if content}(T[t]) \subseteq \bigcap_{j \in Q_1^q} W_j \\ & \text{and card}(W_{g_t}) \leq t; \\ \text{content}(T[t]) \cup W_p \cup \{z\}, & \text{otherwise.} \end{cases}$$

If A is a subset of input language, then go to stage $s+1$, with $q = q+1$ and p unchanged.
4.3 Output a grammar for the language B where:

$$B = \begin{cases} \text{content}(T[t]), & \text{if content}(T[t]) \not\subseteq W_p; \\ \text{content}(T[t]) \cup \{z\}, & \text{if content}(T[t]) \subseteq W_p \\ & \text{card}(W_{g_t}) \leq t; \\ \text{content}(T[t]) \cup W_p \cup \{z\}, & \text{otherwise.} \end{cases}$$

If B is a subset of input language, then go to stage $s+1$, with $p = p+1$ and q unchanged.
Otherwise go to substage $t+1$.
End substage t.
End Stage s

Now suppose a text T for $L \in \mathcal{L}$ is given.

If $L = N$, then clearly $\mathbf{M}'$ will never leave step 1, and for all but finitely many s output a grammar for N.

If $L \neq N$, but $L \in \text{INIT}$, then also $\mathbf{M}'$ will never leave step 1, and for all but finitely many s, output a grammar for L.

Otherwise, $\mathbf{M}$ will eventually get a counterexample in step 1 (as otherwise, $\mathbf{M}'$ will almost always output a grammar for N in step 1, and the input is neither N nor in INIT – eventually leading to a counterexample).

Now, let z be the minimal element which does not belong to L. Note that for all stages s such that the minimal element missing in $T[s]$ is not z, $\mathbf{M}'$ will change stage either due to step 1, step 2, step 3 or step 4.2. Thus, eventually the value of z as computed in step 2 will indeed be the minimal element missing from content(T), and this value will not change thereafter.

We first claim that there are finitely many stages. First note that, after z in the construction achieves its final value, if p achieves a value such that $W_p = L$, it will never change its value (as the conjecture at step 4.1 will not contain a counterexample, and conjecture of B at step 4.3 will produce a counterexample). Thus value of p eventually stabilizes. Furthermore, at every stage after the first counterexample is received in step 1, a change of stage is accompanied by increment in value of either p or q. Thus, we have that either there are finitely many stages or there exists a stage s such that at stage s value of q is such that (i) Q_1^q, Q_2^q are respectively the set of j such that $\mathbf{M}$ asks a question of the form 'is $W_j \supseteq L$' on T and gets yes/no answers where the counterexamples provided are the least ones, and (ii) $\mathbf{M}$ does not ask any questions beyond $T[s]$, and (iii) for each $j \in Q_2^q$, $\min(L - W_j) \in \text{content}(T[s])$, and (iv) for each $j \in Q_2^q$, $\{x \mid x \in L, x < \min(L - W_j)\} \subseteq W_{j,s}$, and (v) $(\forall y < z)[y \in \text{content}(T[s-1])]$ and $(\exists y > z)[y \in \text{content}(T[s-1])]$, and (vi) value of p has achieved its final value before stage s, and $\mathbf{M}'$ has received a counterexample before stage s.

Now we claim that $\mathbf{M}'$ will not go beyond stage s. It has already received a counterexample, so step 1 would not change the stage. Step 2 also does not change the stage by (iii) and (iv) above. At stage s step 3 would not succeed by hypotheses (i) to (v) above. In step 4.2, in each substage, counterexample would be provided for the conjecture A (as z or some value $< z$). Steps 4.1, 4.3 do not change the stage, as p has stabilized.

Thus, let s be the last stage that is executed. Now since step 3 never succeeds, we have that $\mathbf{M}$ will not ask any more questions beyond $T[s]$, and all the answers given to $\mathbf{M}$ on questions asked on prefixes of $T[s]$ in the simulation at step 4 are correct (otherwise either step 3 would succeed, or first clause in the definition of A at step 4.2 would ensure that $\mathbf{M}$ does not get a counterexample in some substage t).

Now if L is finite, then for all but finitely many substages $\text{card}(W_{g_t}) \leq t$, and $\text{content}(T[t]) = L$, and hence $\mathbf{M}'$ would output a conjecture for $L \cup \{z\}$. On the other hand if L is infinite, then for all but finitely many substages t, $\text{card}(W_{g_t}) > t$, and hence $\mathbf{M}'$ would output a conjecture for $W_p \cup \{z\}$. Here note that $W_p \subseteq L$ (as step 4.1 did not produce a counterexample at each substage) and $W_p \supseteq L$ (as at step 4.3, conjecture of B produced a counterexample in each substage).

It follows that $\mathbf{M}'$ eventually outputs conjectures for L or $L \cup \{z\}$. Thus, $\mathbf{M}'$ $\mathbf{BNC}^1\mathbf{Bc}^1$-identifies $\mathcal{L}$. ∎

In fact, the above proof showed that $\bigcup_{m \in N} \mathbf{SupQ}^*\mathbf{Bc}^m(\mathbf{M})$ is contained in $\mathbf{BNCBc}^1(\mathbf{M}')$. Thus, we also have the following.

Theorem 43. $\mathbf{SupQ}^*\mathbf{Ex}^* \subseteq \mathbf{ResBNCBc}^1$.

7.2. *Short negative counterexample versus query models*

Conversely, one 'no' answer, assuming existence of a short counterexample, can sometimes do better than any number queries of any type returning least counterexamples (for the model **LSubQ** we have two different variants of a solution to the problem in question).

Theorem 44 ([10]). *Suppose $n \in N$.*

(a) $\mathbf{ResBNC}^1\mathbf{Ex} - \mathbf{LSubQ}^n\mathbf{Bc}^* \neq \emptyset$.
(b) $\mathbf{ResBNC}^1\mathbf{Bc} - \mathbf{LSubQ}^*\mathbf{Bc}^* \neq \emptyset$.
(c) $\mathbf{ResBNC}^1\mathbf{Ex} - \mathbf{LEquQ}^n\mathbf{Bc}^* \neq \emptyset$.
(d) $\mathbf{ResBNC}^1\mathbf{Ex} - \mathbf{LSupQ}^*\mathbf{Bc}^* \neq \emptyset$.

Proof. Jain and Kinber [10] showed these diagonalizations for $\mathbf{ResNC}^1$ instead of $\mathbf{ResBNC}^1$ above. The proof there also works for $\mathbf{ResBNC}^1$. ∎

(a), (b) above is the strongest possible for diagonalization from **BNC**-model against **SubQ**-model, as $\mathbf{ResSubQ}^*\mathbf{Ex}^a = \mathbf{NCEx}^a = \mathbf{LNCEx}^a$ [10,11] and $\mathbf{LBNCEx}^a \subseteq \mathbf{LNCEx}^a$ [11], and thus, $\mathbf{ResBNCEx}^a \subseteq \mathbf{LBNCEx}^a \subseteq \mathbf{ResSubQ}^*\mathbf{Ex}^a$. Similarly, (c) above is the strongest as $\mathcal{E} \in \mathbf{LEquQ}^*\mathbf{Ex}$ [10].

Acknowledgements

The preliminary version of this paper appeared in COLT' 2006. We thank the anonymous referees of COLT' 2006 and of this journal for several helpful comments. The first author was supported in part by NUS grant numbers R252-000-127-112, R252-000-212-112 and R252-000-308-112.

References

[1] D. Angluin, Queries and concept learning, Machine Learning 2 (1988) 319–342.
[2] J. Bārzdiņš, Two theorems on the limiting synthesis of functions, in: Theory of Algorithms and Programs, vol. 1, Latvian State University, 1974, pp. 82–88 (in Russian).
[3] L. Blum, M. Blum, Toward a mathematical theory of inductive inference, Information and Control 28 (1975) 125–155.

[4] G. Baliga, J. Case, S. Jain, Language learning with some negative information, Journal of Computer and System Sciences 51 (5) (1995) 273–285.
[5] J. Case, C. Lynes, Machine inductive inference and language identification, in: M. Nielsen, E.M. Schmidt (Eds.), Proceedings of the 9th International Colloquium on Automata, Languages and Programming, in: Lecture Notes in Computer Science, vol. 140, Springer-Verlag, 1982, pp. 107–115.
[6] J. Case, C. Smith, Comparison of identification criteria for machine inductive inference, Theoretical Computer Science 25 (1983) 193–220.
[7] M. Fulk, Prudence and other conditions on formal language learning, Information and Computation 85 (1990) 1–11.
[8] W. Gasarch, G. Martin, Bounded Queries in Recursion Theory, Birkhauser, 1998.
[9] E.M. Gold, Language identification in the limit, Information and Control 10 (1967) 447–474.
[10] S. Jain, E. Kinber, Learning languages from positive data and a finite number of queries, Information and Computation 204 (1) (2006) 123–175.
[11] S. Jain, E. Kinber, Learning languages from positive data and negative counterexamples, Journal of Computer and System Sciences (2007) (in press).
[12] S. Jain, D. Osherson, J. Royer, A. Sharma, Systems that Learn: An Introduction to Learning Theory, second ed., MIT Press, Cambridge, MA, 1999.
[13] S. Lange, S. Zilles, Comparison of query learning and Gold-style learning in dependence of the hypothesis space, in: Shai Ben-David, John Case, Akira Maruoka (Eds.), Algorithmic Learning Theory: Fifteenth International Conference, ALT'2004, in: Lecture Notes in Artificial Intelligence, vol. 3244, Springer-Verlag, 2004, pp. 99–113.
[14] S. Lange, S. Zilles, Replacing limit learners with equally powerful one-shot query learners, in: John Shawe-Taylor, Yoram Singer (Eds.), Proceedings of the Seventeenth Annual Conference on Learning Theory, in: Lecture Notes in Artificial Intelligence, vol. 3120, Springer-Verlag, 2004, pp. 155–169.
[15] T. Motoki, Inductive inference from all positive and some negative data, Information Processing Letters 39 (4) (1991) 177–182.
[16] D. Osherson, M. Stob, S. Weinstein, Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists, MIT Press, 1986.
[17] D. Osherson, S. Weinstein, Criteria of language learning, Information and Control 52 (1982) 123–138.
[18] H. Rogers, Theory of Recursive Functions and Effective Computability, McGraw-Hill, 1967, reprinted by MIT Press in 1987.
[19] R. Wiehagen, T. Zeugmann, Ignoring data may be the only way to learn efficiently, Journal of Experimental and Theoretical Artificial Intelligence 6 (1994) 131–144.
[20] T. Zeugmann, S. Lange, A guided tour across the boundaries of learning recursive languages, in: K. Jantke, S. Lange (Eds.), Algorithmic Learning for Knowledge-Based Systems, in: Lecture Notes in Artificial Intelligence, vol. 961, Springer-Verlag, 1995, pp. 190–258.